



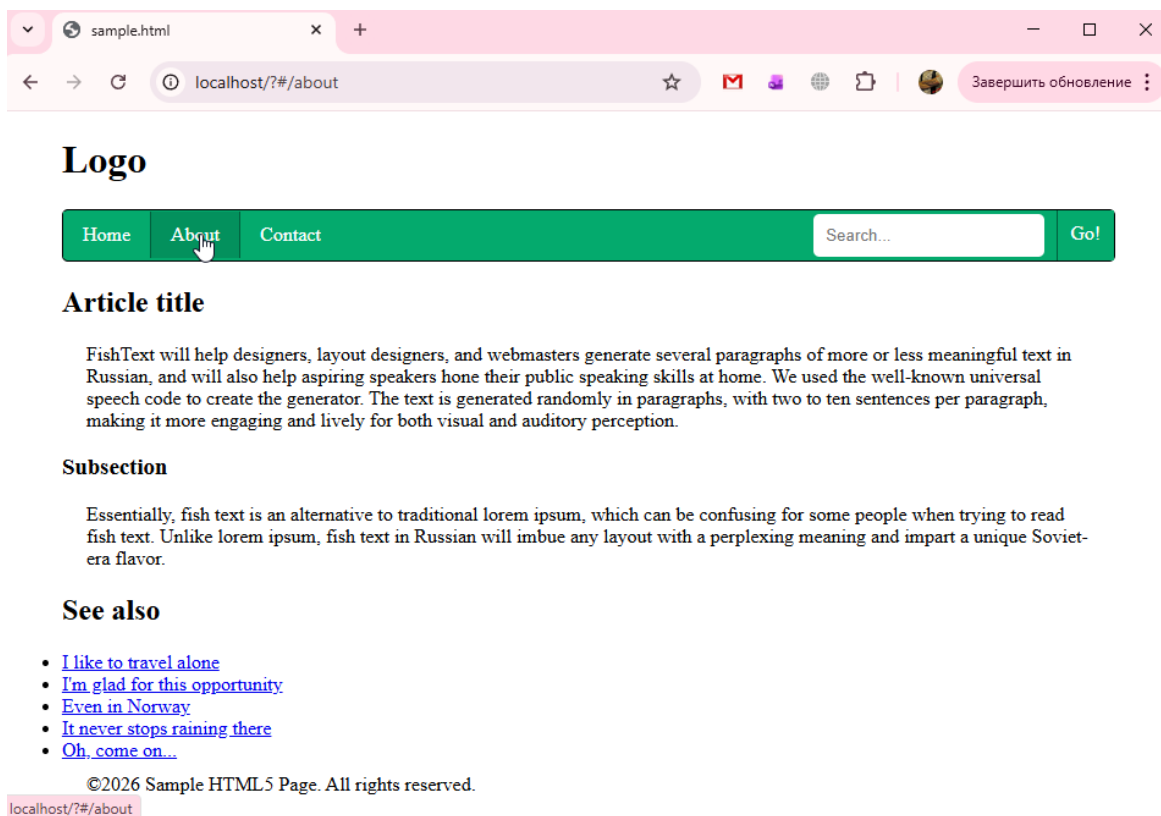
4.3. Файл sample.html

- Глава 4.0. Шаблон HTML и динамические страницы
 - 4.1. Шаблон HTML5
 - 4.2. Шаблон PHP и динамические страницы
 - 4.3. Файл sample.html
 - 4.3.1. Файл style.css
 - 4.3.2. Файл scripts.js
 - 4.4. Файл index.php часть 1
 - 4.5. Файл header.php
 - 4.6. Файл footer.php
 - 4.7. Файл index.php часть 2
 - 4.7.1. Файл home.php
 - 4.7.2. Файл about.php
 - 4.7.3. Файл contact.php
 - 4.7.4. Файл like.php
 - 4.7.5. Файл 404.php

Введение

Пример sample.html

Ниже приведен пример файла sample.html, который является единой статической страницей



Файл sample.html

Ниже приведен код страницы sample.html

[sample.html](#)

```
1. <!doctype html>
2. <html lang="en" xml:lang="en"
   xmlns="http://www.w3.org/1999/xhtml">
3.   <head>
4.     <meta charset="utf-8">
5.     <!-- запрет автоперевода перевода google translate -->
6.     <meta name="google" content="notranslate">
7.     <!-- отображение фактической ширины экрана для адаптивного дизайна -->
8.     <meta name="viewport" content="width=device-width, initial-
   scale=1.0">
9.     <title>sample.html</title>
10.    <link
11.
   href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:
   300|Sonsie+One"
12.      rel="stylesheet"
13.      type="text/css" />
14.    <!-- следующие 3 строки нужны для корректного отображения
```

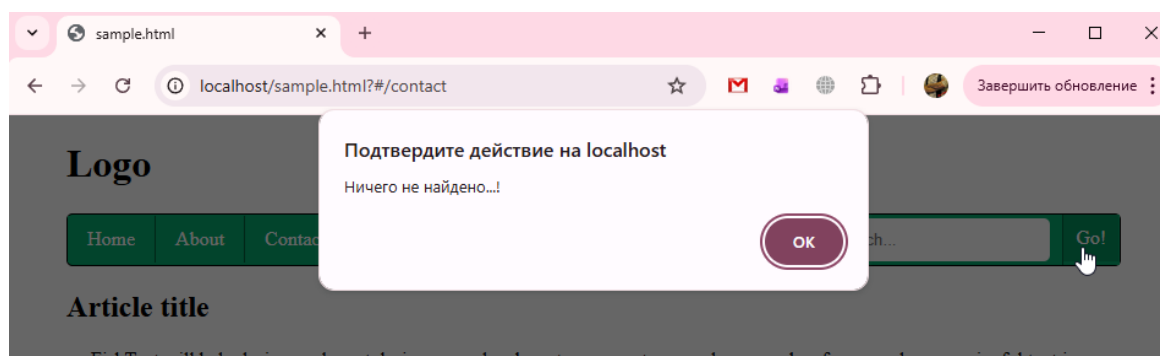
```
семантических элементов HTML5 в старых версиях Internet Explorer-->
15.     <!--[if lt IE 9]>
16.         <script
17.             src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5s
18.             hiv.js"></script>
19.         <![endif]>
20.     <!-- блок стилей CSS -->
21. <style>
22. html {
23.     width: 100%; /* Ширина 90% */
24.     margin: 0;
25.     padding: 0;
26. }
27. body {
28.     width: 90%; /* Задает ширину 90% от вьюпорта */
29.     margin: 0 auto; /* Центрирует контент */
30.     box-sizing: border-box; /* Учитывает padding внутри ширины */
31. }
32. h, p {
33.     margin: 10px;
34.     padding: 0px 10px; /* Убирает внутренние отступы */
35. }
36. .navbar {
37.     display: flex;
38.     justify-content: space-between;
39.     align-items: center;
40.     background-color: #04aa6d;
41.     color: white;
42.     border: 1px solid black;
43.     border-radius: 5px; /* Скругляет все углы на 5px */
44. }
45. ul {
46.     list-style-type: none;
47.     margin: 0;
48.     padding: 0;
49.     overflow: hidden;
50.     background-color: #04aa6d;
51.     display: flex;
52. }
53. ul li {
54.     float: left;
55.     border-right: 1px solid #025f3d;
56. }
57. ul li:last-child {
58.     border-right: none;
59. }
60. ul li a {
61.     display: block;
62.     color: white;
63.     text-align: center;
64.     /* top, right, bottom, left */
```

```
63.     padding: 10px 16px;
64.     text-decoration: none;
65. }
66. ul li a:hover:not(.active) {
67.     background-color: #03915d;
68. }
69. ul li a.active {
70.     background-color: #04aa6d;
71. }
72. .search-form {
73.     display: flex;
74. }
75. .search-form input {
76.     background-color: #D3D3D3
77.     border: 1px solid #025f3d;
78.     border-radius: 5px; /* Скругляет все углы на 5px */
79.     /* top, right, bottom, left */
80.     margin: 3px 10px 3px 0; /* 10px отступ справа */
81.     padding: 10px 10px; /* Убирает внутренние отступы */
82.     border: none;
83. }
84. .search-form-button {
85.     /* top, right, bottom, left */
86.     margin: 0 1px 0 0; /* 10px отступ справа */
87.     padding: 10px 10px; /* Убирает внутренние отступы */
88.     border: none; /* Убирает рамку (если есть) */
89.     background: none; /* Убирает фоновый цвет */
90.     color: white;
91.     font: inherit; /* Наследует шрифт родителя */
92.     cursor: pointer; /* Возвращает курсор-ручку */
93. }
94. ul li .search-form-button:hover:not(.active) {
95.     background-color: #03915d;
96. }
97. ul li .search-form-button.active {
98.     background-color: #04aa6d;
99. }
100. </style>
101. </head>
102. <body>
103.     <!-- Вот наш главный заголовок, который используется на всех страницах
        нашего веб-сайта -->
104.     <header>
105.         <h1>Logo</h1>
106.     </header>
107. <nav class="navbar">
108.     <ul>
109.         <ul>
110.             <li><a class="active" href="#home">Home</a></li>
111.             <li><a href="#/about">About</a></li>
```

```
112.         <li><a href="#/contact">Contact</a></li>
113.     </ul>
114. </ul>
115. <!-- Форма поиска. -->
116. <form class="search-form">
117.     <ul>
118.         <li><input type="search" placeholder="Search..."></li>
119.         <li><button class = "search-form-button" type="submit"
120.             id="searchButton" value="Go!">Go!</button></li>
121.     </ul>
122. </form>
122. </nav>
123. <!-- Здесь основное содержимое нашей страницы -->
124. <main>
125.     <!-- Она содержит статью -->
126.     <article>
127.         <h2>Article title</h2>
128.         <p>
129.             FishText will help designers, layout designers, and
130.             webmasters generate
131.             several paragraphs of more or less meaningful text in
132.             Russian, and will
133.             also help aspiring speakers hone their public speaking
134.             skills at home.
135.             We used the well-known universal speech code to create
136.             the generator.
137.             The text is generated randomly in paragraphs, with two
138.             to ten sentences
139.             per paragraph, making it more engaging and lively for
140.             both visual and
141.             auditory perception.
142.         </p>
143.         <h3>Subsection</h3>
144.         <p>
145.             Essentially, fish text is an alternative to
146.             traditional lorem ipsum,
147.             which can be confusing for some people when trying to
148.             read fish text.
149.             Unlike lorem ipsum, fish text in Russian will imbue
150.             any layout with a
151.             perplexing meaning and impart a unique Soviet-era
152.             flavor.
153.         </p>
154.     </article>
155.     <!-- Дополнительный контент также может быть вложен в основной
156.         контент -->
157.     <aside>
158.         <h2>See also</h2>
159.         <li><a href="#">I like to travel alone</a></li>
160.         <li><a href="#">I'm glad for this opportunity</a></li>
161.         <li><a href="#">Even in Norway</a></li>
```

```
151.         <li><a href="#">It never stops raining there</a></li>
152.         <li><a href="#">Oh, come on...</a></li>
153.     </aside>
154. </main>
155. <!-- И вот наш главный нижний колонтитул, который используется на всех
    страницах нашего веб-сайта -->
156. <footer>
157.     <p>©2026 Sample HTML5 Page. All rights reserved.</p>
158. </footer>
159. <!-- Optional: Link to a JavaScript file -->
160. <script>
161. // Находим кнопку по ID
162.     const button = document.getElementById('searchButton');
163. // Добавляем обработчик события "click"
164.     button.addEventListener('click', function() {
165. // Выводим alert
166.         alert('Ничего не найдено...!');
167.     });
168. </script>
169. </body>
170. </html>
```

Не стоит пугаться, так как данная страница просто пример HTML страницы со стилями и небольшим примером работы языка JavaScript для кнопки поиска «Go!»



Хотя, нет, стоит...

Большенство просто пролистает этот код, так как его очень не удобно читать. В примере все заключенно в тело одной страницы и для создания дополнительных страниц, необходимо полностью копировать весь текст в создаваемые страницы, и уже там выискивать и менять текст в нужных блоках.

Это может быть приемлемо только для маленьких сайтов-визиток, где контент долгое время не меняется.

А если текст меняется постоянно? Если на страницах разные стили оформления? Если у гостей и зарегистрированных пользователей ограничения в отображаемых пунктах меню?

Тут на помощь приходят серверные языки программирования (PHP, Python, Ruby, Node.js), базы данных (MySQL, PostgreSQL) и клиентский JavaScript, которые совместно генерируют контент в реальном времени. Они позволяют обрабатывать запросы пользователей, взаимодействовать с

сервером и изменять интерфейс без перезагрузки.

Но, все по порядку...

Для начала вынесем стили и скрипты в отдельные директории и файлы, далее разделим страницу на части и следом с помощью PHP подключим все это в единый динамический шаблон.

И начнем мы с файла **style.css**, описанного на следующей странице руководства

Дополнения и Файлы

см. также:

- [Консорциум Всемирной паутины \(W3C\)](#)
- [Объявление языка в HTML](#)

🔗 4.2 Шаблон PHP и динамические страницы 4.3.1. Файл style.css 🔗

From:
<https://wwoss.direct.quickconnect.to/> - worldwide open-source software

Permanent link:
https://wwoss.direct.quickconnect.to/doku.php?id=software:development:demo:cms:ucms:sample_php_page_sample

Last update: 2026/02/09 18:19

