

Таблицы

Модуль 5 (2 пары)

Создание таблицы

- Таблица - набор данных, распределенных по строкам и ячейкам. В большинстве ячеек размещаются табличные данные, остальные содержат заголовки для строк и столбцов, описывающие содержимое.
- Для создания таблицы в HTML-документе используется тег `<table>`, он представляет собой контейнер, в котором находится все содержимое таблицы.
- Создание таблицы всегда начинается со строк, которые определяются с помощью тега `<tr>`, каждая строка, в свою очередь, состоит из ячеек. Тег `<tr>` может содержать в себе только теги для создания ячеек.
- В HTML существует два разных тега для создания ячеек, первым из них является `<td>`, он создает обычные ячейки с данными. По умолчанию, содержимое тегов `<td>` выравнивается по левому краю. Второй тег для создания ячеек - это тег `<th>`, он позволяет определять ячейки, которые содержат заголовки для столбцов или строк, содержимое таких ячеек отображается жирным текстом и выравнивается по центру. Теги `<td>` и `<th>` могут включать в себя любые HTML-элементы, которые доступны для использования в теле документа.

```
<table border="1">
  <tr><th>Первый заголовок</th><th>Второй заголовок</th></tr>
  <tr><td>строка 1, ячейка 1</td><td>строка 1, ячейка 2</td></tr>
  <tr><td>строка 2, ячейка 1</td><td>строка 2, ячейка 2</td></tr>
</table>
```

Первый заголовок	Второй заголовок
строка 1, ячейка 1	строка 1, ячейка 2
строка 2, ячейка 1	строка 2, ячейка 2

Таблица внутри таблицы

В HTML есть возможность создания вложенных таблиц, то есть таких таблиц, которые расположены внутри других таблиц. Чтобы сделать вложенную таблицу надо код таблицы, которую требуется сделать вложенной, поместить внутрь любого тега `<td>`.

Для примера, возьмём уже созданную ранее нами таблицу и поместим этот код во вторую ячейку второй строки:

```
<table border="1">
  <tr><th>Первый заголовок</th><th>Второй заголовок</th></tr>
  <tr><td>строка 1, ячейка 1</td><td>строка 1, ячейка 2</td></tr>
  <tr><td>строка 2, ячейка 1</td>
    <td>строка 2, ячейка 2
      <table border="1">
        <tr><th>Первый заголовок</th><th>Второй заголовок</th></tr>
        <tr><td>строка 1, ячейка 1</td><td>строка 1, ячейка 2</td></tr>
        <tr><td>строка 2, ячейка 1</td><td>строка 2, ячейка 2</td></tr>
      </table>
    </td>
  </tr>
</table>
```

Первый заголовок	Второй заголовок	
строка 1, ячейка 1	строка 1, ячейка 2	
строка 2, ячейка 1	строка 2, ячейка 2	
	Первый заголовок	Второй заголовок
	строка 1, ячейка 1	строка 1, ячейка 2
	строка 2, ячейка 1	строка 2, ячейка 2

Рамка таблицы

- Тег `<table>` может иметь необязательный атрибут `border`, управляющий границами таблицы. По умолчанию браузеры не отображают рамку вокруг таблицы и ячеек, но присутствие атрибута `border`, указывает браузеру, что у таблицы должна быть рамка.
- По стандарту HTML5 значением атрибута `border` может быть либо 1, либо пустая строка (обозначается как пара двойных кавычек ""). Не важно, какое значение атрибута вы зададите, если он присутствует, то рамка будет отображаться:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Заголовок страницы</title>
  </head>
  <body>
    <h4>Таблица с рамкой:</h4>
    <table border="1">
      <tr><td>Первая</td><td>строка</td></tr>
      <tr><td>Вторая</td><td>строка</td></tr>
    </table>
    <h4>Таблица без рамки:</h4>
    <table>
      <tr><td>Первая</td><td>строка</td></tr>
      <tr><td>Вторая</td><td>строка</td></tr>
    </table>
  </body>
</html>
```

Таблица с рамкой:

Первая	строка
Вторая	строка

Таблица без рамки:

Первая строка
Вторая строка

Рамка таблицы

- Если вам не нравится вид рамки у таблицы, заданный по умолчанию, т.е. двойные рамки, вы можете воспользоваться атрибутом `style`, прописав в нем CSS свойство `border-collapse` со значением `collapse`, это придаст рамке стандартный вид:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Заголовок страницы</title>
  </head>
  <body>

    <table border="1" style="border-collapse: collapse;">
      <tr><td>ячейка 1</td><td>ячейка 2</td></tr>
      <tr><td>ячейка 3</td><td>ячейка 4</td></tr>
    </table>

  </body>
</html>
```

ячейка 1	ячейка 2
ячейка 3	ячейка 4

Заголовок таблицы

- У каждой таблицы обычно бывает заголовок, описывающий ее содержимое. Для создания заголовка таблицы используется тег `<caption>`, обычно он размещается сразу после открывающего тега `<table>`, но также он может располагаться практически в любом месте внутри таблицы между элементами `<tr>`.
- Тег заголовка может содержать в себе любые элементы, которые можно использовать в теле HTML-документа. По умолчанию содержимое тега `<caption>` отображается над таблицей и выравнивается по ее центру. Расположение заголовка можно изменить применив к нему CSS свойство `caption-side`:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Заголовок страницы</title>
  </head>
  <body>

    <table border="1">
      <caption>Моя первая таблица</caption>
      <tr><td>строка 1, ячейка 1</td><td>строка 1, ячейка 2</td></tr>
      <tr><td>строка 2, ячейка 1</td><td>строка 2, ячейка 2</td></tr>
    </table>

  </body>
</html>
```

Моя первая таблица

строка 1, ячейка 1	строка 1, ячейка 2
строка 2, ячейка 1	строка 2, ячейка 2

Объединение ячеек в таблице

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Заголовок страницы</title>
  </head>
  <body>
    <h4>Объединим две ячейки в строке таблицы:</h4>
    <table border="1">
      <tr>
        <th>Имя</th><th colspan="2">Телефон</th>
      </tr>
      <tr>
        <td>Петя Иванов</td><td>123 45 67</td><td>234 56 78</td>
      </tr>
    </table>
    <h4>Объединим две ячейки в столбце таблицы:</h4>
    <table border="1">
      <tr>
        <th>Имя</th><td>Петя Иванов</td>
      </tr>
      <tr>
        <th rowspan="2">Телефон</th><td>123 45 67</td>
      </tr>
      <tr>
        <td>234 56 78</td></tr>
    </table>
  </body>
</html>
```

Объединим две ячейки в строке таблицы: Объединим две ячейки в столбце таблицы:

Имя	Телефон	
Петя Иванов	123 45 67	234 56 78

Имя	Петя Иванов
Телефон	123 45 67
	234 56 78

- Смежные ячейки в таблице могут объединяться по горизонтали или вертикали с целью размещения большего объема данных. Например, в таблице из четырех строк и четырех столбцов все ячейки в первой строке можно объединить и поместить в этой строке красивый заголовок. Возможно также объединение нескольких строк или столбцов.

- Для объединения ячеек по горизонтали, то есть объединение идущих друг за другом ячеек в строке таблицы, используется атрибут `colspan`, в значении задается число смежных ячеек в строке, которое вы хотите объединить.

- Для объединения ячеек по вертикали, то есть объединение расположенных друг под другом ячеек в столбце таблицы, используется атрибут `rowspan`, в значении задается число смежных ячеек столбца, которое вы хотите объединить.

Объединение ячеек в таблице

- Атрибуты `colspan` и `rowspan` объединяют ячейки таблицы. Атрибут `colspan` задает количество ячеек, объединенных по горизонтали, а `rowspan` — по вертикали.

```
<table>
  <tr>
    <th>№ п/п</th>
    <th>Наименование товара</th>
    <th>Ед. изм.</th>
    <th>Количество</th>
    <th>Цена за ед. изм., руб.</th>
    <th>Стоимость, руб.</th>
  </tr>
  <tr>
    <td>1.</td>
    <td>Томаты свежие</td><td>кг</td><td>15,20</td><td>69,00</td><td>1048,80</td>
  </tr>
  <tr>
    <td>2.</td>
    <td>Огурцы свежие</td><td>кг</td><td>2,50</td><td>48,00</td><td>120,00</td>
  </tr>
  <tr>
    <td colspan="5">ИТОГО:</td><td>1168,80</td><!-- Задаем количество
ячеек по горизонтали для объединения-->
  </tr>
</table>
```

№ п/п	Наименование товара	Ед. изм.	Количество	Цена за ед. изм., руб.	Стоимость, руб.
1.	Томаты свежие	кг	15,20	69,00	1048,80
2.	Огурцы свежие	кг	2,50	48,00	120,00
ИТОГО:					1168,80

Атрибуты элементов таблицы

- Атрибуты `colspan` и `rowspan` объединяют ячейки таблицы. Атрибут `colspan` задает количество ячеек, объединенных по горизонтали, а `rowspan` — по вертикали.

Атрибут	Описание, принимаемое значение
<code>colspan</code>	<p>Количество ячеек в строке для объединения по горизонтали.</p> <pre><td colspan="3"></pre> <p>Возможные значения: число от 1 до 999.</p>
<code>headers</code>	<p>Задаёт список ячеек заголовка, содержащих информацию о заголовке текущей ячейки данных. Предназначен для речевых браузеров.</p> <pre><th id="идентификатор">...</th></pre> <pre><th headers="идентификатор">...</th></pre> <p>Принимаемые значения: список имен ячеек, разделенных пробелами; эти имена должны быть присвоены ячейкам через их атрибут <code>id</code>.</p>
<code>rowspan</code>	<p>Количество ячеек в столбце для объединения по вертикали.</p> <pre><td rowspan="2"></pre> <p>Возможные значения: число от 1 до 999.</p>
<code>span</code>	<p>Количество колонок, объединяемых для задания единого стиля, по умолчанию равно 1.</p> <pre><col span="2"></pre> <p>Принимаемые значения: любое целое положительное число.</p>

CSS: Оформление таблиц

Рамка таблицы

- По умолчанию HTML таблица на веб-странице отображается без рамки, для добавления рамки к таблице, как и ко всем другим элементам, используется CSS свойство `border`. Но стоит обратить внимание на то, что если добавить рамку только к элементу `<table>`, то она отобразится вокруг всей таблицы. Для того, чтобы ячейки таблицы тоже имели рамку, надо будет установить свойство `border` и для элементов `<th>` и `<td>`.

```
table, th, td { border: 1px solid black; }
```

Имя	Фамилия
Гомер	Симпсон
Мардж	Симпсон

- Теперь и таблица и ячейки имеют рамки, при этом и каждая ячейка и таблица имеют свои собственные рамки. В результате между рамками появилось пустое пространство, управлять размером этого пространства позволяет свойство [border-spacing](#), которое задается для всей таблицы целиком. Другими словами, нельзя управлять промежутками между различными ячейками индивидуально.

Рамка таблицы

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      table, td, th {
        border: 4px outset blue;
        border-spacing: 5px;
      }
      .first { border-collapse: collapse; }
    </style>
  </head>

  <body>
    <table>
      <tr><th>Имя</th><th>Фамилия</th></tr>
      <tr><td>Гомер</td><td>Симпсон</td></tr>
      <tr><td>Мардж</td><td>Симпсон</td></tr>
    </table>
    <br>
    <table class="first">
      <tr><th>Имя</th><th>Фамилия</th></tr>
      <tr><td>Гомер</td><td>Симпсон</td></tr>
      <tr><td>Мардж</td><td>Симпсон</td></tr>
    </table>
  </body>
</html>
```

Имя	Фамилия
Гомер	Симпсон
Мардж	Симпсон

- Даже если убрать промежутки между ячейками с помощью значения 0 свойства **border-spacing**, то рамки ячеек будут соприкасаться друг с другом, удваиваясь. Для объединения рамок ячеек используется свойство **border-collapse**. Оно может принимать два значения:
 - separate**: является значением по умолчанию. Ячейки отображаются на небольшом расстоянии друг от друга, каждая ячейка имеет свою собственную рамку.
 - collapse**: соединяет соседние рамки в одну, все промежутки между ячейками, а также между ячейками и рамкой таблицы игнорируются.

Имя	Фамилия
Гомер	Симпсон
Мардж	Симпсон

Размер таблицы

- После добавления рамок к ячейкам таблицы стало заметно, что содержимое ячеек слишком близко расположено к краям. Для добавления свободного пространства между краями ячеек и их содержимым можно воспользоваться свойством `padding`:

```
th, td { padding: 7px; }
```

- Размер таблицы зависит от ее содержимого, но часто возникают ситуации, когда таблица оказывается слишком узкой и появляется необходимость ее растянуть. Ширину и высоту таблицы можно изменять с помощью свойств `width` и `height`, задавая нужные размеры или самой таблице или ячейкам:

```
table { width: 70%; }  
th { height: 50px; }
```

Имя	Фамилия
Гомер	Симпсон
Мардж	Симпсон

Выравнивание текста

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      table, td, th {
        border: 1px solid black;
        border-collapse: collapse;
      }
      table { width: 70% }
      td { text-align: right; }
      th { height: 50px; }
      .test1 { vertical-align: top; }
      .test2 { vertical-align: bottom; }
    </style>
  </head>
  <body>
    <table>
      <tr><th class="test1">Имя</th><th
class="test2">Фамилия</th></tr>
      <tr><td>Гомер</td><td>Симпсон</td></tr>
      <tr><td>Мардж</td><td>Симпсон</td></tr>
    </table>
  </body>
</html>
```

- По умолчанию текст в заголовочных ячейках таблицы выравнивается по центру, а в обычных ячейках текст выровнен по левому краю, используя свойство **text-align** можно управлять выравниванием текста по горизонтали.

- CSS свойство **vertical-align** позволяет управлять выравниванием текстового содержимого по вертикали. По умолчанию текст выровнен вертикально по центру ячеек. Вертикальное выравнивание можно переопределить с помощью одного из значений свойства vertical-align:

- top:** текст выравнивается по верхней границе ячейки
- middle:** выравнивает текст по центру (значение по умолчанию)
- bottom:** текст выравнивается по нижней границе ячейки

Имя	Фамилия
Гомер	Симпсон
Мардж	Симпсон

Чередование фонового цвета строк таблицы

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      table {
        width: 70%;
        border-collapse: collapse;
      }
      td, th {
        border: 1px solid #98bf21;
        padding: 3px 7px 2px 7px;
      }
      th {
        text-align: left;
        padding: 5px;
        background-color: #A7C942;
        color: #fff;
      }
      .alt td { background-color: #EAF2D3; }
    </style>
  </head>
  <body>
    <table>
      <tr><th>Имя</th><th>Фамилия</th><th>Положение</th></tr>
      <tr><td>Гомер</td><td>Симпсон</td><td>отец</td></tr>
      <tr class="alt"><td>Мардж</td><td>Симпсон</td><td>мать</td></tr>
      <tr><td>Барт</td><td>Симпсон</td><td>сын</td></tr>
      <tr class="alt"><td>Лиза</td><td>Симпсон</td><td>дочь</td></tr>
    </table>
  </body>
</html>
```

- При просмотре больших таблиц, содержащих много строк с большим количеством информации, бывает трудно отследить, какие данные относятся к конкретной строке. Чтобы помочь пользователям сориентироваться, можно использовать два разных фоновых цвета поочередно. Для создания описанного эффекта можно использовать селектор class, добавляя его к каждой второй строке таблицы:

Имя	Фамилия	Положение
Гомер	Симпсон	отец
Мардж	Симпсон	мать
Барт	Симпсон	сын
Лиза	Симпсон	дочь

Чередование фонового цвета строк таблицы

- Добавлять атрибут `class` к каждой второй строке довольно утомительное занятие. В CSS3 был добавлен псевдо-класс `:nth-child`, позволяющий решить эту проблему альтернативным путем. Теперь эффекта чередования можно достичь исключительно средствами CSS, не прибегая к изменению HTML-разметки документа. С помощью псевдо-класса `:nth-child` можно выбрать все четные или нечетные строки таблицы, используя одно из ключевых слов: `even` (четные) или `odd` (нечетные):

```
tr:nth-child(odd) { background-color: #EAF2D3; }
```


Изменение фона строки при наведении курсора

- Еще одним способом повышения удобочитаемости табличных данных является изменение фонового цвета строки при наведении на нее курсора мыши. Это поможет выделить нужное содержимое таблицы и повысит визуальное восприятие данных.
- Реализовать такой эффект очень просто, для этого нужно добавить псевдо-класс **:hover** к селектору строки таблицы и задать нужный цвет фона:

```
tr:hover { background-color: #E0E0FF; }
```

Имя	Фамилия	Положение
Гомер	Симпсон	отец
Мардж	Симпсон	мать
Барт	Симпсон	сын
Лиза	Симпсон	дочь

Выравнивание таблицы по центру

Выравнивание HTML таблицы по центру возможно только в том случае, если ширина таблицы меньше, чем ширина её родительского элемента. Чтобы выровнять таблицу по центру, надо воспользоваться свойством `margin`, задав ему минимум два значения: первое значение будет отвечать за внешний отступ таблицы сверху и снизу, а второе - за автоматическое выравнивание по центру:

```
table { margin: 10px auto; }
```

Имя	Фамилия
Гомер	Симпсон
Мардж	Симпсон

Если вам нужны разные отступы сверху и снизу таблицы, то можно задать свойству `margin` три значения: первое будет отвечать за отступ сверху, второе за выравнивание по горизонтали, а третье за отступ снизу:

```
table { margin: 10px auto 30px; }
```

Пример создания таблицы

* Меню ресторана "Ромашка" *

Кухня	Холодные блюда		Горячие блюда		Десерты
	Салаты	Закуски	Первые блюда	Вторые блюда	
Русская	Винегрет	Язык с хреном	Щи с квашеной капустой	Вареники с картошкой	Печеные яблоки с медом
	Оливье	Студень говяжий	Рассольник домашний	Караси запеченные в сметане	Блинчатый пирог
	Сельдь под "шубой"	Судак заливной	Мясная солянка	Котлеты "Пожарские"	Пирожное "Картошка"
Испанская	Севиче из гребешков	Эмпанадас	Хлебный суп с чесноком	Пазлья с морепродуктами	Чуррос
	Тимбал из авокадо и тунца	Ахотомате	Астурийская фабада	Свиное раксо	Альмойшавена
	Фасоль с ветчиной	Чанфайна	Рыбный суп с манными клецками	Тортилья картофельная	Бунуэлос
Французская	Вогезский салат	Рийет из курицы	Баклажанный крем-суп "Ренуар"	Картофель огратен	Бриоши
	Салат "Панзанелла"	Делисьез из сыра	Французский тыквенный суп	Гратин из птицы	Лигурийский лимонный пирог
	Тар-тар	Маринованный лосось	Суп "Конти"	Тартифлетт	Саварен "Триумф"

Пример создания таблицы

```

<table>
  <caption>Меню ресторана "Ромашка"</caption>
  <tr>
    <th rowspan="2" class="first">Кухня</th>
    <th colspan="2">Холодные блюда</th>
    <th colspan="2">Горячие блюда</th>
    <th rowspan="2">Десерты</th>
  </tr>
  <tr>
    <td class="first">Салаты</td>
    <td class="first">Закуски</td>
    <td class="first">Первые блюда</td>
    <td class="first">Вторые блюда</td>
  </tr>
  <tr>
    <td rowspan="3" class="first">Русская</td>
    <td>Винегрет</td>
    <td>Язык с хреном</td>
    <td>Щи с квашеной капустой</td>
    <td>Вареники с картошкой</td>
    <td>Печеные яблоки с медом</td>
  </tr>
  <tr>
    <td>Оливье</td>
    <td>Студень говяжий</td>
    <td>Рассольник домашний</td>
    <td>Караси запеченные в сметане</td>
    <td>Блинчатый пирог</td>
  </tr>
  <tr>
    <td>Сельдь под "шубой"</td>
    <td>Судак заливной</td>
    <td>Мясная солянка</td>
    <td>Котлеты "Пожарские"</td>
    <td>Пирожное "Картошка"</td>
  </tr>

```

Пример создания таблицы

```
<tr>
  <td rowspan="3" class="first">Испанская</td>
  <td>Севиче из гребешков</td>
  <td>Эмпанадас</td>
  <td>Хлебный суп с чесноком</td>
  <td>Паэлья с морепродуктами</td>
  <td>Чуррос</td>
</tr>
<tr>
  <td>Тимбал из авокадо и тунца</td>
  <td>Ахотомате</td>
  <td>Астурийская фабада</td>
  <td>Свиное раксо</td>
  <td>Альмойшавена</td>
</tr>
<tr>
  <td>Фасоль с ветчиной</td>
  <td>Чанфайна</td>
  <td>Рыбный суп с манными клецками </td>
  <td>Тортилья картофельная</td>
  <td>Бунуэлос</td>
</tr>
<tr>
  <td rowspan="3" class="first">Французская</td>
  <td>Вогезский салат</td>
  <td>Рийет из курицы</td>
  <td>Баклажанный крем-суп "Ренуар"</td>
  <td>Картофель огратен</td>
  <td>Бриоши</td>
</tr>
```

```
<tr>
  <td>Салат "Панзанелла"</td>
  <td>Делисьез из сыра</td>
  <td>Французский тыквенный суп</td>
  <td>Гратин из птицы</td>
  <td>Лигурийский лимонный пирог</td>
</tr>
<tr>
  <td>Тап-тап</td>
  <td>Маринованный лосось</td>
  <td>Суп "Конти"</td>
  <td>Тартифлетт</td>
  <td>Саварен "Триумф"</td>
</tr>
</table>
```

Пример создания таблицы

```
body {
  margin: 0;
  background: #F4F1F8;
}
table {
  border-collapse: collapse;
  line-height: 1.1;
  font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
  background: radial-gradient(farthest-corner at 50% 50%, white, #DCECF8);
  color: #0C213B;
}
caption {
  font-family: annabelle, cursive;
  font-weight: bold;
  font-size: 2em;
  padding: 10px;
  color: #F3CD26;
  text-shadow: 1px 1px 0 rgba(0,0,0,.3);
}
caption:before, caption:after {
  content: "\274B";
  color: #A9E2CC;
  margin: 0 10px;
}

th {
  padding: 10px;
  border: 1px solid #A9E2CC;
}
td {
  font-size: 0.8em;
  padding: 5px 7px;
  border: 1px solid #A9E2CC;
}
.first {
  font-size: 1em;
  font-weight: bold;
  text-align: center;
}
```

Позиционирование

Понятие позиционирования

- Позиционирование - это управление местом расположения (позицией) элемента на веб-странице, оно контролируется с помощью CSS свойства `position`.
- Для указания точного места расположения позиционированных элементов, используются CSS свойства: `top`, `right`, `bottom` и `left`. Они работают со всеми позиционированными элементами, кроме статических.

Элементы могут перекрывать друг друга!

Пример позиционирования.

Отображение элемента над остальными!

- Свойство `position` имеет 4 значения: **static**, **fixed**, **relative** и **absolute**. Каждое из этих значений будет продемонстрировано ниже с примером использования.
- Перед тем как подробно разобрать все виды позиционирования элементов на странице, нам придется рассмотреть, что такое поток документа.

Поток документа

По умолчанию элементы на веб-странице отображаются в том порядке, в котором они представлены в HTML-документе, т. е. блочные элементы занимают всю доступную для них ширину и укладываются вертикально один под другим. Строчные элементы выстраиваются по горизонтали до тех пор, пока не будет занята вся доступная ширина, после того как вся ширина будет занята, будет сделан перенос строки и всё пойдет по новой. Такой порядок расположения элементов называется **нормальным потоком** (его также называют **потоком документа** или **общим потоком**).

С помощью свойства **float** или **position** можно убрать элемент из нормального потока. Если элемент "выпадает" из нормального потока, то элементы, которые расположены в коде ниже этого элемента будут смещены на его место на веб-странице.

Схемы позиционирования

- В CSS блок элемента может быть расположен в соответствии с тремя схемами позиционирования:
 - **Нормальный поток**
 - Нормальный поток включает блочный контекст форматирования (элементы с **display block**, **list-item** или **table**), строчный (встроенный) контекст форматирования (элементы с **display inline**, **inline-block** или **inline-table**), и относительное и «липкое» позиционирование элементов уровня блока и строки.
 - **Обтекание**
 - В обтекающей модели блок удаляется из нормального потока и позиционируется влево или вправо. Содержимое обтекает правую сторону элемента с **float: left** и левую сторону элемента с **float: right**.
 - **Абсолютное позиционирование**
 - В модели абсолютного позиционирования блок полностью удаляется из нормального потока и ему присваивается позиция относительно содержащего блока. Абсолютное позиционирование реализуется с помощью значений **position: absolute**; и **position: fixed**;
 - Элементом «вне потока» может быть плавающий, абсолютно позиционированный или корневой элемент.

Содержащий блок

- Положение и размер блока(ов) элемента иногда вычисляются относительно некоторого прямоугольника, называемого содержащим блоком элемента (containing block). В общих словах, содержащий блок — это блок, который содержит другой элемент. В случае нормального потока корневой элемент html является содержащим блоком для элемента body, который, в свою очередь, является содержащим блоком для всех его дочерних элементов и так далее. В случае позиционирования содержащий блок полностью зависит от типа позиционирования. Содержащий блок элемента определяется следующим образом:
- Содержащий блок, в котором находится корневой элемент, представляет собой прямоугольник — так называемый начальный содержащий блок.
- Для некорневого элемента с `position: static`; или `position: relative`; содержащий блок формируется краем области содержимого ближайшего родительского блока уровня блока, ячейки таблицы или уровня строки.
- Содержащим блоком элемента с `position: fixed`; является окно просмотра.
- Для некорневого элемента с `position: absolute`; содержащим блоком устанавливается ближайший родительский элемент со значением `position: absolute/relative/fixed` следующим образом:
 - — если предок — элемент уровня блока, содержащим блоком будет область содержимого плюс поля элемента `padding`;
 - — если предок — элемент уровня строки, содержащим блоком будет область содержимого;
 - — если предков нет, то содержащий блок элемента определяется как начальный содержащий блок.
- Для «липкого» блока содержащим блоком является ближайший предок с прокруткой или окно просмотра, в противном случае.

Выбор схемы позиционирования: свойство position

- Свойство position определяет, какой из алгоритмов позиционирования используется для вычисления положения блока.
- Свойство не наследуется.
- **Static** - Блок располагается в соответствии с нормальным потоком. Свойства top, right, bottom и left не применяются. Значение по умолчанию.
- **Initial** - Устанавливает значение свойства в значение по умолчанию.
- **Inherit** - Наследует значение свойства от родительского элемента.
- **Sticky** - Положение блока рассчитывается в соответствии с нормальным потоком. Затем блок смещается относительно своего ближайшего предка с прокруткой или окна просмотра, если ни у одного из предков нет прокрутки.
- «Липкий» блок может перекрывать другие блоки, а также создавать полосы прокрутки в случае переполнения.
- «Липкий» блок сохраняет свои размеры, включая разрывы строк и пространство, первоначально зарезервированное для него.
- «Липкий» блок создает новый содержащий блок для абсолютно и относительно позиционированных потомков.

Выбор схемы позиционирования: свойство `position`

- **Relative** - Положение блока рассчитывается в соответствии с нормальным потоком. Затем блок смещается относительно его нормального положения и во всех случаях, включая элементы таблицы, не влияет на положение любых следующих блоков. Тем не менее, такое смещение может привести к перекрытию блоков, а также к появлению полосы прокрутки в случае переполнения.
- Относительно позиционированный блок сохраняет свои размеры, включая разрывы строк и пространство, первоначально зарезервированное для него.
- Относительно позиционированный блок создает новый содержащий блок для абсолютно позиционированных потомков.
- Влияние `position: relative;` на элементы таблицы определяется следующим образом:
- Элементы с `table-row-group`, `table-header-group`, `table-footer-group` и `table-row` смещаются относительно их обычной позиции в таблице. Если ячейки таблицы занимают несколько строк, смещаются только ячейки начальной строки.
- `table-column-group`, `table-column` не смещает соответствующий столбец и не оказывает визуального влияния.
- `table-caption` and `table-cell` смещаются относительно своего нормального положения в таблице. Если ячейка таблицы охватывает несколько столбцов или строк, то она смещается целиком.

Выбор схемы позиционирования: свойство `position`

- **Absolute** - Положение блока (и, возможно, размер) задается с помощью свойств `top`, `right`, `bottom` и `left`. Эти свойства определяют явное смещение относительно его содержащего блока. Абсолютно позиционированные блоки полностью удаляются из нормального потока, не влияя на расположение сестринских элементов.
- Отступы `margin` абсолютно позиционированных блоков не схлопываются.
- Абсолютно позиционированный блок создает новый содержащий блок для дочерних элементов нормального потока и потомков с `position: absolute`;
- Содержимое абсолютно позиционированного элемента не может обтекать другие блоки. Абсолютно позиционированный блок могут скрывать содержимое другого блока (или сами могут быть скрыты), в зависимости от значения `z-index` перекрывающихся блоков.
- **Fixed** - Фиксированное позиционирование аналогично абсолютному позиционированию, с отличием в том, что для содержащим блоком устанавливается окно просмотра. Такой блок полностью удаляется из потока документа и не имеет позиции относительно какой-либо части документа. Фиксированные блоки не перемещаются при прокрутке документа. В этом отношении они похожи на фиксированные фоновые изображения.
- При печати фиксированные блоки повторяются на каждой странице, содержащим блоком для них устанавливается область страницы. Блоки с фиксированным положением, которые больше области страницы, обрезаются.

Выбор схемы позиционирования: свойство position

- Синтаксис

```
position: static;  
position: relative;  
position: absolute;  
position: sticky;  
position: fixed;  
position: initial;  
position: inherit;
```

Заголовок {position: absolute; top: 10px;}

Заголовок {position: static}

Заголовок с **position static** сохраняет своё место в потоке документа

Заголовок {position: relative; top: 10px;}

Заголовок с **position relative** смещается относительно своего текущего положения в макете

Заголовок с **position absolute** смещается относительно верхнего левого угла браузера. При этом его ширина становится равной ширине содержимого

Смещение блока: свойства **top**, **right**, **bottom**, **left**

- Элемент считается позиционированным, если свойство **position** имеет значение, отличное от **static**. Позиционированные элементы генерируют позиционированные блоки и могут быть расположены в соответствии со следующими четырьмя физическими свойствами **top**, **right**, **bottom**, **left**.
- Свойство **top** задает расстояние, на которое верхний край абсолютно позиционированного блока (с учетом его **margin**) смещается ниже верхнего края содержащего блока. Для относительно позиционированных блоков определяет смещение относительно верхнего края самого блока (то есть блоку задается позиция в нормальном потоке, а затем смещение от этой позиции в соответствии с этим свойством).

Значение	Описание
auto	Влияние значения зависит типа элемента. Значение по умолчанию.
длина	Смещение на фиксированном расстоянии от указанного края. Отрицательные значения допускаются.
%	Процентные значения вычисляются относительно высоты содержащего блока. Для «липкого» блока — относительно высоты корневого элемента. Отрицательные значения допускаются.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Смещение блока: свойства **top**, **right**, **bottom**, **left**

- Свойство **right** указывает расстояние, на которое правый край абсолютно позиционированного блока (с учетом его **margin**) смещен влево от правого края содержащего блока. Для относительно позиционированных блоков определяет смещение относительно правого края самого блока.

Значение	Описание
auto	Влияние значения зависит типа элемента. Значение по умолчанию.
длина	Смещение на фиксированном расстоянии от указанного края. Отрицательные значения допускаются.
%	Процентные значения вычисляются относительно ширины содержащего блока. Для «липкого» блока — относительно ширины корневого элемента. Отрицательные значения допускаются.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Смещение блока: свойства **top**, **right**, **bottom**, **left**

- Свойство **bottom** указывает расстояние, на которое нижний край блока смещен вверх относительно нижнего края содержащего блока. Для относительно позиционированных блоков определяет смещение относительно нижнего края самого блока.

Значение	Описание
auto	Влияние значения зависит типа элемента. Значение по умолчанию.
длина	Смещение на фиксированном расстоянии от указанного края. Отрицательные значения допускаются.
%	Процентные значения вычисляются относительно высоты содержащего блока. Для «липкого» блока — относительно высоты корневого элемента. Отрицательные значения допускаются.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Смещение блока: свойства **top**, **right**, **bottom**, **left**

- Свойство **left** указывает расстояние, на которое левый край смещен вправо от левого края содержащего блока. Для относительно позиционированных блоков определяет смещение относительно левого края самого блока.
- Положительные значения смещают элемент внутрь содержащего блока, а отрицательные — за его пределы.

Значение	Описание
auto	Влияние значения зависит типа элемента. Значение по умолчанию.
длина	Смещение на фиксированном расстоянии от указанного края. Отрицательные значения допускаются.
%	Процентные значения вычисляются относительно ширины содержащего блока. Для «липкого» блока — относительно ширины корневого элемента. Отрицательные значения допускаются.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

Статическое позиционирование

- Статическим называют позиционирование, заданное по умолчанию для всех элементов на веб-странице. Если к элементу не применять свойство `position`, то он будет статическим и будет отображаться на веб-странице согласно общему потоку элементов.
- При применении CSS свойств `top`, `left`, `right` или `bottom` к статически позиционированному элементу, они будут игнорироваться.
- Если есть необходимость, то можно задать статическое позиционирование в таблице стилей с помощью значения `static`:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      p { position: static; }
    </style>
  </head>

  <body>
    <p>Первый абзац.</p>
    <p>Второй абзац.</p>
  </body>
</html>
```

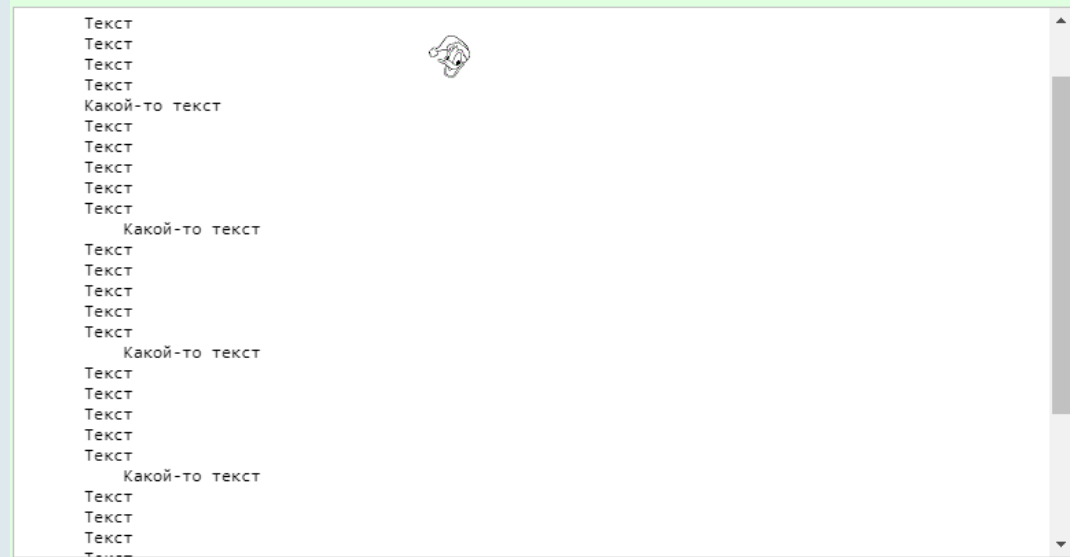
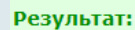
Первый абзац.

Второй абзац.

Фиксированное позиционирование

- Элементы с фиксированным позиционированием располагаются на странице относительно окна браузера. Такие элементы удаляются из общего потока, элементы, следующие в потоке за фиксированным элементом, будут игнорировать его, смещаясь и занимая его место на веб-странице.
- Стоит обратить внимание на то, что элементы с фиксированным позиционированием могут перекрывать другие элементы, скрывая их полностью или частично. При прокрутке длинных страниц, они создают эффект неподвижных объектов, оставаясь на одном и том же месте:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      img {
        position: fixed;
        top: 5%;
        left: 40%
      }
    </style>
  </head>
  <body>
    
    <pre>
      Текст
      Текст
      .....
      Текст
      Текст
      Текст
    </pre>
  </body>
</html>
```



Относительное позиционирование

- Элементы с относительным позиционированием, как и статические элементы, остаются в общем потоке. При применении свойств `top`, `left`, `right` или `bottom` к относительно позиционированным элементам, они будут смещаться относительно своего местоположения, оставляя пустое пространство там, где элемент располагался изначально.
- Такие элементы не влияют на расположение окружающих их элементов, остальные элементы остаются на своих местах и могут быть перекрыты относительно позиционированным элементом.
- Примечание: элементы с относительным позиционированием (`relative`) обычно используют в качестве родителя для элементов с абсолютным позиционированием (`absolute`).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      h2 {
        position: relative;
        top: 30px;
      }
    </style>
  </head>

  <body>
    <h1>Заголовок первого уровня.</h1>
    <h2>Относительно позиционированный заголовок.</h2>
    <h3>Заголовок третьего уровня.</h3>
  </body>
</html>
```

Заголовок первого уровня.

Относительно позиционированный заголовок.
Заголовок третьего уровня.

Абсолютное позиционирование

- Элементы с абсолютным позиционированием полностью удаляются из общего потока, остальные элементы будут занимать освободившееся пространство, полностью игнорируя абсолютно позиционированные элементы. Затем можно позиционировать элемент в любое нужное место веб-страницы, используя свойства `top`, `left`, `right` или `bottom`.
- Все абсолютно позиционированные элементы размещаются относительно окна браузера или относительно ближайшего позиционированного предка (если он есть), у которого свойство `position` имеет значение `absolute`, `fixed` или `relative`.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      img {
        position: absolute;
        right: 0px;
        top: 0px;
      }
    </style>
  </head>
```

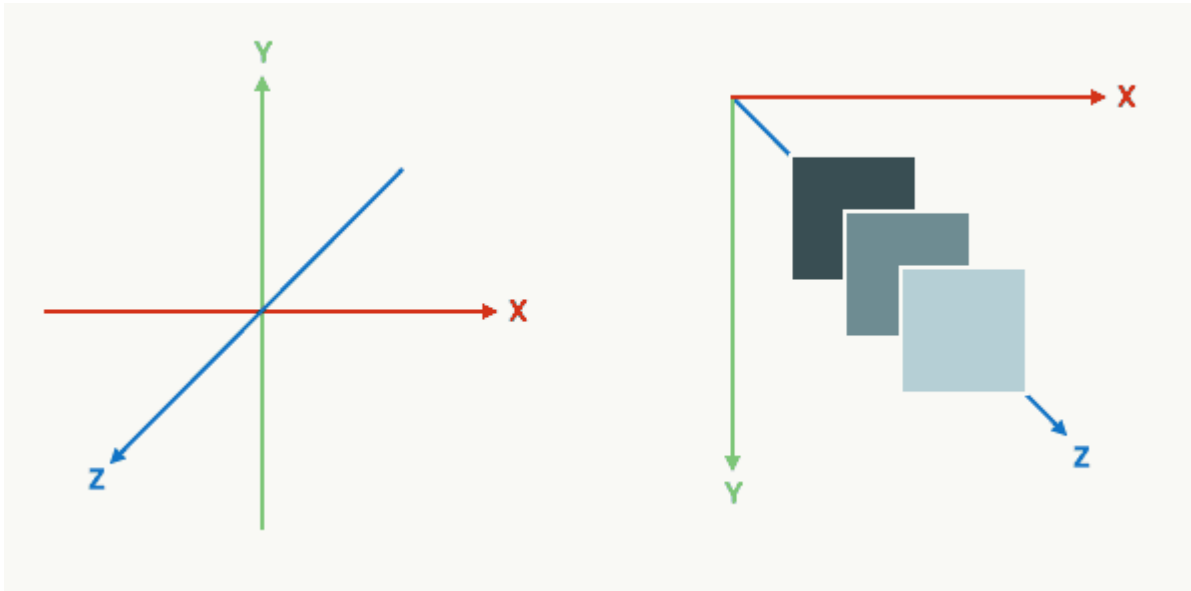
Изменим расположение логотипа с помощью абсолютного позиционирования.
Теперь логотип будет располагаться в правом верхнем углу страницы.

```
<body>
  
  <p>Изменим расположение логотипа с помощью абсолютного позиционирования.<br>
  Теперь логотип будет располагаться в правом верхнем углу страницы.</p>
</body>
</html>
```



Определение контекста наложения: свойство z-index

- В CSS каждый блок имеет позицию в трех измерениях. В дополнение к горизонтальному и вертикальному положению, блоки выкладываются вдоль оси Z друг над другом. Положение вдоль оси Z особенно важно, когда блоки визуально накладываются друг на друга.



- Порядок, в котором дерево документа отрисовывается на экране, описывается с помощью контекста наложения. Каждый блок принадлежит одному контексту наложения. Каждый блок в данном контексте наложения имеет целочисленный уровень, который является его положением на оси Z относительно других блоков в том же контексте наложения.

Определение контекста наложения: свойство z-index

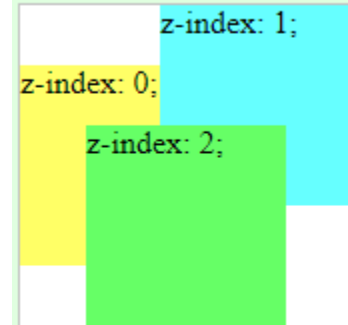
- Блоки с более высокими уровнями всегда отображаются перед блоками с более низкими уровнями, а блоки с одинаковым уровнем располагаются снизу вверх в соответствии с порядком следования элементов в исходном документе. Блок элемента имеет ту же позицию, что и блок его родителя, если только ему не присвоен другой уровень свойством z-index.
- Свойство z-index позволяет изменить способ наложения элементов друг на друга.
- Свойство не наследуется.

Значение	Описание
auto	Вычисляется в 0. Если для блока задано position: fixed; или это корневой элемент, значение auto также устанавливает новый контекст наложения. Значение по умолчанию.
целое число	Определяет положение блока в текущем контексте наложения. Также устанавливает новый локальный контекст наложения. Можно использовать любое целое число, включая отрицательные числа. Отрицательные значения помещают элемент вглубь экрана.
inherit	Наследует значение свойства от родительского элемента.
initial	Устанавливает значение свойства в значение по умолчанию.

Определение контекста наложения: свойство z-index

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      div {
        position: absolute;
        width: 100px;
        height: 100px;
      }
      .div1 {
        background-color: #66FFFF;
        left: 70px;
        top: 0px;
        z-index: 1;
      }
      .div2 {
        background-color: #FFFF66;
        left: 0px;
        top: 30px;
        z-index: 0;
      }
      .div3 {
        background-color: #66FF66;
        left: 33px;
        top: 60px;
        z-index: 2;
      }
    </style>
  </head>
  <body>
    <div class="div1">z-index: 1;</div>
    <div class="div2">z-index: 0;</div>
    <div class="div3">z-index: 2;</div>
  </body>
</html>
```

Результат:



Гиперссылки

HTML-ссылки

- HTML-ссылки создаются с помощью элементов `<a>`, `<area>` и `<link>`. Ссылки представляют собой связь между двумя ресурсами, одним из которых является текущий документ.
- Ссылки можно поделить на две категории:
- **ссылки на внешние ресурсы** — создаются с помощью тега `<link>` и используются для расширения возможностей текущего документа при обработке браузером;
- **гиперссылки** — ссылки на другие ресурсы, которые пользователь может посетить или загрузить.

Структура ссылки

Гиперссылки создаются с помощью парного тега `<a>`. Внутри тега помещается текст, который будет отображаться на веб-странице. Текст ссылки отображается в браузере с подчёркиванием, цвет шрифта — синий, при наведении на ссылку курсор мыши меняет вид.

Обязательным параметром тега `<a>` является атрибут `href`, который задает URL-адрес веб-страницы.

```
<a href="http://site.ua">указатель ссылки</a>
```

Ссылка состоит из двух частей — указателя и адресной части. Указатель ссылки представляет собой фрагмент текста или изображение, видимые для пользователя. Адресная часть ссылки пользователю не видна, она представляет собой адрес ресурса, к которому необходимо перейти.

Адресная часть ссылки состоит из URL. URL (Uniform Resource Locator) — унифицированный адрес ресурса. При создании адресов для разделения слов между собой рекомендуется использовать дефис, а не символ подчёркивания. В общем виде URL имеющий следующий формат:

```
метод доступа://имя сервера:порт/путь
```

Метод доступа

- Метод доступа, или протокол, осуществляет обмен данными между рабочими станциями в разных сетях. Наиболее распространенные протоколы передачи данных:

- **file** обеспечивает чтение файла с локального диска:

file:/gallery/pictures/summer.html

- **http** предоставляет доступ к веб-странице по протоколу HTTP:

http://site.ua/

- **https** — специальная реализация протокола HTTP, использующая шифрование (как правило, SSL или TLS)

https://site.ua/

- **ftp** осуществляет запрос к FTP-серверу на получение файла:

ftp://pgu/directory/library

- **mailto** запускает сеанс почтовой связи с указанным адресатом и хостом:

mailto:serhii.ruban81@gmail.com

Метод доступа

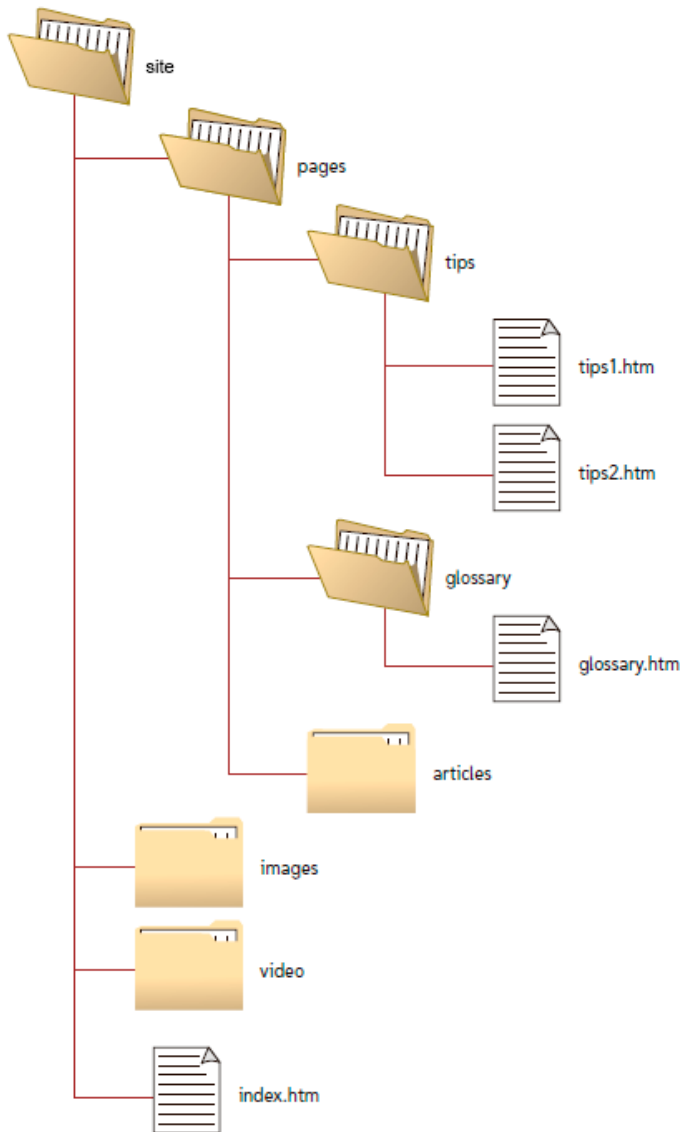
- Имя сервера описывает полное имя машины в сети, например, site.ru. Если имя сервера не указано, то ссылка считается локальной, т.е. она относится к той же машине, на которой находится HTML-документ, содержащий ссылку.

- Номер порта TCP, на котором функционирует веб-сервер. Представляет собой число, которое необходимо указывать, если метод требует номер порта (отдельные сервера могут иметь свой отличительный номер порта). Если порт не указан, по умолчанию используется порт 80. Стандартными портами являются:

- 21 — FTP
- 23 — Telnet
- 70 — Gopher
- 80 — HTTP

- Путь содержит имя папки, в которой находится файл.

Абсолютный и относительный путь



- Когда в ссылке указывается только имя файла, браузер предполагает, что файл находится в той же папке, что и документ, содержащий гиперссылку. На практике веб-сайты содержат сотни документов, которые размещают в отдельные папки, чтобы ими было легче управлять. Чтобы создать ссылку на файл, находящийся вне папки, содержащей текущий документ, необходимо указать расположение файла или путь. HTML поддерживает два вида пути: абсолютный и относительный.

Абсолютный путь

- Абсолютный путь указывает точное местоположение файла в пределах всей структуры папок на компьютере (сервере). Абсолютный путь к файлу даёт доступ к файлу со сторонних ресурсов и содержит следующие компоненты:
 - 1) протокол, например, http (опционально);
 - 2) домен (доменное имя или IP-адрес компьютера);
 - 3) папка (имя папки, указывающей путь к файлу);
 - 4) файл (имя файла).
- Существует два вида записи абсолютного пути — с указанием протокола и без него:

```
http://site.ru/pages/tips/tips1.html  
//site.ru/pages/tips/tips1.html
```

- Если файл находится в корневой папке, то путь к файлу будет следующим:

```
http://site.ru/index.html
```

- При отсутствии имени файла будет загружаться веб-страница, которая задана по умолчанию в настройках веб-сервера (так называемый индексный файл).

```
http://site.ru/
```

- Обычно в качестве индексного файла выступает документ с именем index.html. Наличие завершающего слеша / означает, что обращение идет к папке, если его нет — напрямую к файлу.

Относительный путь

- Относительный путь описывает путь к указанному документу относительно текущего. Путь определяется с учётом местоположения веб-страницы, на которой находится ссылка. Относительные ссылки используются при создании ссылок на другие документы на одном и том же сайте. Когда браузер не находит в ссылке протокол `http://`, он выполняет поиск указанного документа на том же сервере.
- Относительный путь содержит следующие компоненты:
 - 1) папка (имя папки, указывающей путь к файлу);
 - 2) файл (имя файла).
- Путь для относительных ссылок имеет три специальных обозначения:
 - `/` указывает на корневую директорию и говорит о том, что нужно начать путь от корневого каталога документов и идти вниз до следующей папки
 - `./` указывает на текущую папку
 - `../` подняться на одну папку (директорию) выше
- Главное отличие относительного пути от абсолютного в том, что относительный путь не содержит имени корневой папки и родительских папок, что делает адрес короче, и в случае переезда с одного домена на другой не нужно прописывать новый абсолютный адрес. Но если сторонний ресурс будет ссылаться например, на ваши изображения с относительными адресами, то они не будут отображаться на другом сайте.

Якоря

- Якоря, или внутренние ссылки, создают переходы на различные разделы текущей веб-страницы, позволяя быстро перемещаться между разделами. Это оказывается очень удобным в случае, когда на странице слишком много текста. Внутренние ссылки также создаются при помощи тега `<a>` с разницей в том, что атрибут `href` содержит имя указателя — так называемый якорь, а не URL-адрес. Перед именем указателя всегда ставится знак `#`.
- Следующая разметка создаст оглавление с быстрыми переходами на соответствующие разделы:

```
<h1>Времена года</h1>
<h2>Оглавление</h2>
<a href="#p1">Лето</a> <!--создаём якорь, указав #id элемента-->
<a href="#p2">Осень</a>
<a href="#p3">Зима</a>
<a href="#p4">Весна</a>
<p id="p1">...</p> <!--добавляем соответствующий id элементу-->
<p id="p2">...</p>
<p id="p3">...</p>
<p id="p4">...</p>
```

- Если нужно сделать ссылку **с одной страницы** сайта на определенный раздел **другой страницы**, то необходимо задать `id` для этого раздела страницы, а затем добавить его к абсолютному адресу ссылки:

```
<th id="about-color">color</th>
<a href="https://mysite.ua/css-shrifty/#about-color" class="site"
target="_blank">color</a>
```

Изображение как ссылка

- Чтобы сделать кликабельное изображение, необходимо поместить элемент `` внутри тега `<a>`.

```
<!DOCTYPE html>
<html>
<body>

<h1>HTML Изображение ссылка</h1>
<p>Изображение является ссылкой. Вы можете нажать на нее.</p>
<a href="index.php">
  
</a>
<p>Мы добавили "border:0" в IE9 (и более ранних версиях), чтобы предотвратить
отображение границы вокруг изображения.</p>
</body>
</html>
```

HTML Изображение ссылка

Изображение является ссылкой. Вы можете нажать на нее.



Мы добавили "border:0" в IE9 (и более ранних версиях), чтобы предотвратить отображение границы вокруг изображения.

Атрибут target

- По умолчанию ресурсы, на которые ведут ссылки, открываются в том же окне. С помощью атрибута target можно переопределить это действие. Атрибут target может принимать следующие значения:
- `_blank`: открытие html-документа в новом окне или вкладке браузера
- `_self`: открытие html-документа в том же фрейме (или окне)
- `_parent`: открытие документа в родительском фрейме, если ссылка расположена во внутреннем фрейме
- `_top`: открытие html-документа на все окно браузера
- `frameName`: открытие html-документа во фрейме, который называется `frameName` (В данном случае `frameName` - только пример, название фрейма может быть произвольным)
- Например, открытие документа по ссылке в новом окне:

```
<a href="https://www.rubhoz.com/prognoz-kleva.html" target="_blank">Прогноз клёва  
рыбы в Кривом Роге</a>
```

Стилизация ссылок

- По умолчанию ссылка уже имеет некоторый цвет (один из оттенков синего), кроме того она имеет подчеркивание. При нажатии на ссылку она становится активной и приобретает красный цвет, а после перехода по ссылке эта ссылка может окраситься в другой цвет (как правило, в фиолетовый). Подобная стилизация задается многими браузерами по умолчанию, но мы можем ее переопределить. Например:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ссылки</title>
    <style>
      a:link      {color:blue; text-decoration:none}
      a:visited  {color:pink; text-decoration:none}
      a:hover    {color:red; text-decoration:underline}
      a:active   {color:yellow; text-decoration:underline}
    </style>
  </head>
  <body>
    <a href="index.html">Карта рыбных мест</a>
  </body>
</html>
```

- Здесь определены стили для ссылок в различных состояниях. **a:link** применяется для ссылок в обычном состоянии, когда они не нажаты и на них не наведен указатель мыши.
- a:visited** указывает на состояние ссылки, по которой уже был осуществлен переход.
- a:hover** указывает на состояние ссылки, на которую навели указатель мыши.
- a:active** указывает на ссылку в нажатом состоянии.
- Стиль **color** устанавливает цвет ссылки. А стиль **text-decoration** устанавливает подчеркивание: если значение **underline**, то ссылка подчеркнута, если **none**, то подчеркивание отсутствует.

Подчеркивание ссылок

- **Удаление подчеркивания:**

```
a {text-decoration: none;}
```

- **Добавление подчеркивания только при наведении на ссылку:**

```
a {text-decoration: none;}  
a:hover {text-decoration: underline;}
```

- **Внешний вид нижней границы ссылки:**

```
a {  
  text-decoration: none;  
  border-bottom: 2px dashed DarkOrchid;  
  padding-bottom: 3px;  
}
```

Изображения для ссылок

- Добавить изображение для ссылки можно с помощью CSS-свойства `background-image`. Так как элемент `<a>` является строчным а `{display: inline;}`, то предварительно его нужно преобразовать в блочный элемент а `{display: block;}`.
- Чтобы вставить изображение или иконку перед ссылкой, необходимо добавить отступ с помощью свойства `padding-left`. Этот прием может пригодиться в случае, когда на странице есть ссылки для загрузки каких-либо документов различных форматов, и вы можете добавить значок-изображение типа файла для большей наглядности.
- Если нужно, чтобы значок автоматически добавился ко всем ссылкам, содержащим документы одного формата, можно воспользоваться следующей конструкцией:

```
a[href$=".pdf"] {background-image: url(images/pdf.png);}
```

- Символ `href$` в селекторе атрибута дает браузеру команду найти все атрибуты `href`, заканчивающиеся определенным образом (в данном случае `.pdf`) и добавить к ссылке соответствующий значок.

Как сделать ссылку на телефонный номер, скайп или адрес электронной почты

У ссылок появились новые возможности — по клику можно не только переходить на другие страницы и скачивать файлы, но и совершать звонки на телефоны, отправлять сообщения или звонить по скайпу.

ссылка на телефонный номер

```
<a href="tel:+380978265298">+38 (097) 826-52-98</a>
```

ссылка на адрес электронной почты

```
<a href="mailto:example@gmail.com">example@gmail.com</a>
```

ссылка на скайп (позвонить)

```
<a href="skype:имя-пользователя?call">Skype</a>
```

ссылка на скайп (открыть чат)

```
<a href="skype:имя-пользователя?chat">Skype</a>
```

ссылка на скайп (добавить в список контактов)

```
<a href="skype:имя-пользователя?add">Skype</a>
```

ссылка на скайп (отправить файл)

```
<a href="skype:имя-пользователя?sendfile">Skype</a>
```

Навигация по сайту

Навигация по сайту

Навигация по сайту должна быть удобной. Любой пользователь, «заблудившись» на сайте или не сумев найти необходимую информацию из-за запутанных переходов, просто уйдет к конкурентам.

Система навигации на сайте— это набор гиперссылок, созданный для переходов по разделам сайта для поиска конкретной информации. Ссылки могут быть двух типов: текстовыми и графическими. Отдельно выделяют ссылки-роловеры, которые при нажатии или наведении курсора изменяют вид (размер шрифта, цвет).

Ссылки в тексте должны быть заметными, поэтому принято выделять их с помощью подчеркивания и другого цвета. На активной ссылке при наведении курсора он принимает вид руки.

С точки зрения юзабилити, удобная навигация по сайту должна конкретно отвечать на 3 главных вопроса:

На какой сейчас странице находится посетитель?

Какие страницы он уже посетил?

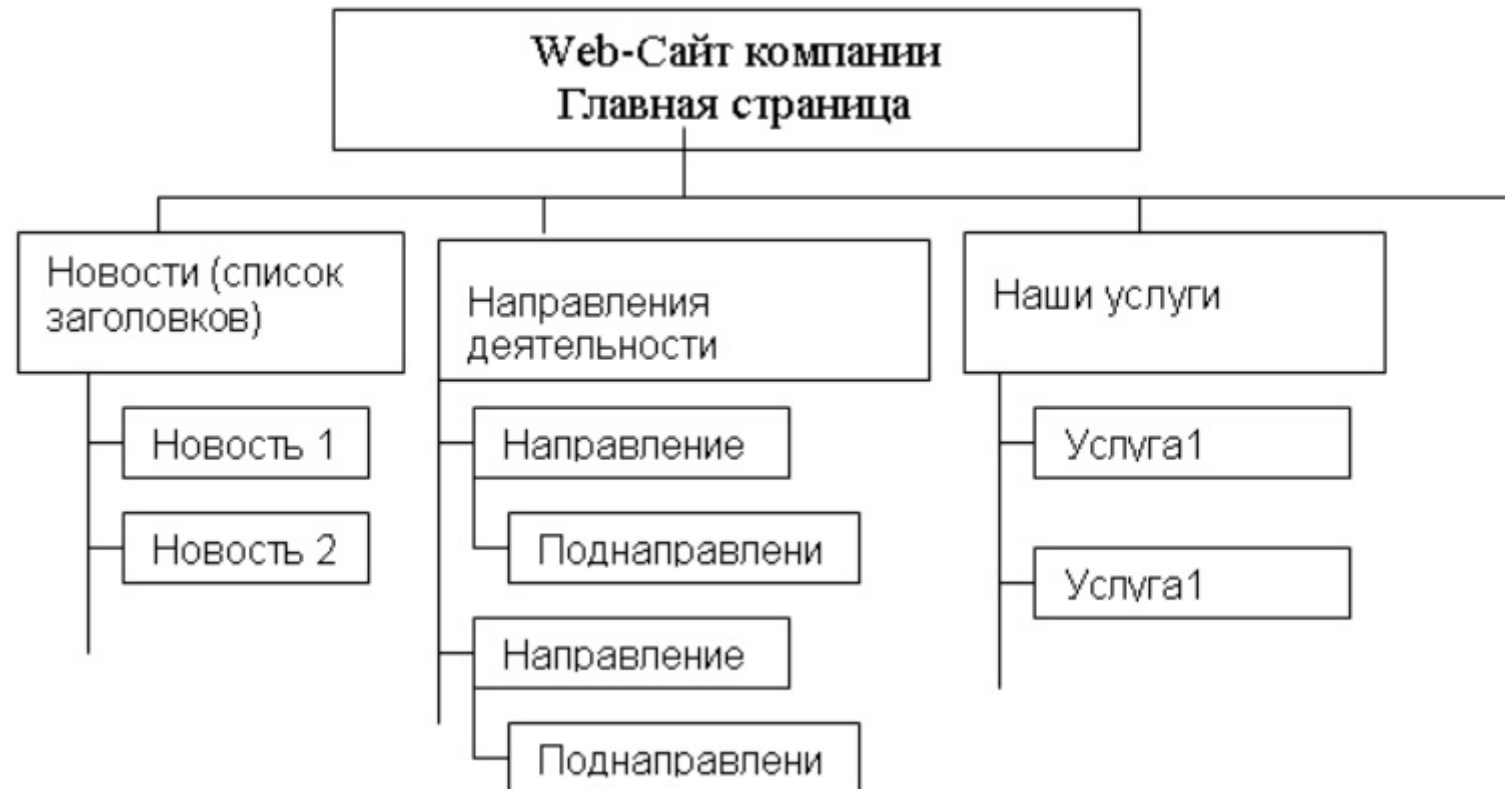
Какие еще страницы ему доступны?

Для ответа на них также существует несколько правил. Ссылки на страницы, которые пользователь уже видел, отличаются по цвету от тех, по которым он еще не переходил.

Многих интересует вопрос: а сколько же ссылок должно быть на сайте? Здесь однозначно ответить нельзя, так как это во многом зависит от количества страниц сайта. А нужное количество определяется лишь опытным путем. Главное правило: их должно быть столько, сколько будет удобно для пользователя. В то же время система навигации не должна занимать слишком много пространства сайта. В среднем ее площадь это 5-15% всего пространства ресурса.

Разработка системы навигации и ее структуры

Работа над гиперссылками и навигацией начинается лишь после того, как определены все задачи сайта и объем контента (текста и графики). В противном случае материалы могут просто не вписаться в разработанную структуру. Если же страница вскоре будет удалена, то и ссылки, связывающие ее с остальными, необходимо будет изменять.



Меню

- Если ваш веб-сайт не ограничивается одной веб-страницей, то стоит подумать о добавлении панели навигации (меню). Меню - раздел веб-сайта, предназначенный помогать посетителю перемещаться по сайту. Любое меню представляет собой список ссылок, ведущих на внутренние страницы сайта. Самым простым способом добавить панель навигации на сайт является создание меню с помощью CSS и HTML.
- По умолчанию все элементы списка `...` располагаются вертикально, занимая по ширине всю ширину элемента контейнера `...`, который в свою очередь занимает всю ширину его блока-контейнера.
- HTML разметка для горизонтальной навигации

```
<ul>
<li><a href="">Пункт меню</a></li>
<li><a href="">Пункт меню</a></li>
<li><a href="">Пункт меню</a></li>
...
</ul>
```

- Горизонтальное меню, находящееся внутри тега `...`, можно дополнительно помещать внутрь элемента `<nav>...</nav>` и/или `<header>...</header>`. Благодаря этому html-разметке придается семантический смысл, а также появляется дополнительная возможность для форматирования блока меню.

Горизонтальное меню

- Существует несколько способов разместить их горизонтально. Для начала нужно сбросить стили браузера по умолчанию для элементов навигации:

```
ul {  
  list-style: none; /*убираем маркеры списка*/  
  margin: 0; /*убираем верхнее и нижнее поле, равное 1em*/  
  padding-left: 0; /*убираем левый отступ, равный 40px*/  
}  
a {text-decoration: none; /*убираем подчеркивание текста  
ссылки*/}
```

1. `li {display: inline;}`



2. `li {float: left;}`



3. `li {display: inline-block;}`



4. `ul {display: flex;}`



Горизонтальное меню

- Способ 1. `li {display: inline;}`
- Делаем элементы списка строчными. В результате они располагаются по горизонтали, с правой стороны между ними добавляется промежуток, равный `0.4em` (вычисляется относительно размера шрифта). Чтобы убрать его, добавляем для `li` отрицательное правое поле `li {margin-right: -4px;}`. Далее стилизуем ссылки по своему желанию.
- Способ 2. `li {float: left;}`
- Делаем элементы списка плавающими. В результате они располагаются по горизонтали. Высота блока-контейнера `ul` становится равной нулю. Чтобы решить эту проблему, добавляем для `ul` `{overflow: hidden;}`, расширяя его и позволяя ему таким образом содержать плавающие элементы. Для ссылок добавляем `a {display: block;}` и стилизуем их по своему желанию.
- Способ 3. `li {display: inline-block;}`
- Делаем элементы списка строчно-блочными. Они располагаются по горизонтали, с правой стороны образуется промежуток, как и в первом случае. Для ссылок добавляем `a {display: block;}` и стилизуем их по своему желанию.
- Способ 4. `ul {display: flex;}`
- Делаем список `ul` гибким контейнером с помощью модели CSS3 `flexbox`. В результате элементы списка располагаются горизонтально. Добавляем для ссылок `a {display: block;}` и стилизуем их по своему желанию.

Горизонтальное меню

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      #navbar {
        margin: 0;
        padding: 0;
        list-style-type: none;
        border: 2px solid #0066FF;
        border-radius: 20px 5px;
        width: 550px;
        text-align: center;
        background-color: #33ADFF;
      }
      #navbar li { display: inline; }
      #navbar a {
        color: #fff;
        padding: 5px 10px;
        text-decoration: none;
        font-weight: bold;
        display: inline-block;
        width: 100px;
      }
      #navbar a:hover {
        border-radius: 20px 5px;
        background-color: #0066FF;
      }
    </style>
  </head>
```

Главная

Новости

Контакты

О нас

```
<body>

  <ul id="navbar">
    <li><a href="#">Главная</a></li>
    <li><a href="#">Новости</a></li>
    <li><a href="#">Контакты</a></li>
    <li><a href="#">О нас</a></li>
  </ul>

</body>
</html>
```


Вертикальное меню

- Первым шагом создания вертикального меню будет создание маркированного списка. Также нам нужно будет иметь возможность идентифицировать список, поэтому мы добавим к нему атрибут `id` с идентификатором `"navbar"`. Каждый элемент `` нашего списка будет содержать по одной ссылке:

```
<ul id="navbar">
  <li><a href="#">Главная</a></li>
  <li><a href="#">Новости</a></li>
  <li><a href="#">Контакты</a></li>
  <li><a href="#">О нас</a></li>
</ul>
```

- Сбрасываем стили списка, установленных по умолчанию. Нам нужно убрать внешние и внутренние отступы у самого списка и маркеры у пунктов списка. Затем зададим нужную ширину:

```
#navbar {
  margin: 0;
  padding: 0;
  list-style-type: none;
  width: 100px;
}
```

Вертикальное меню

- Мы добавим к ним фоновый цвет, изменим параметры текста: цвет, размер и насыщенность шрифта, уберем подчеркивание, добавим небольшие отступы и переопределим отображение элемента `<a>` со строчного на блочный. Дополнительно были добавлены левая и нижняя рамки к пунктам списка.
- Самой важной частью наших изменений является переопределение строчных элементов на блочные. Теперь наши ссылки занимают все доступное пространство пунктов списка, то есть для перехода по ссылке нам больше не нужно наводить курсор точно на текст.

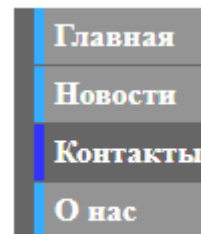
```
#navbar a {
  background-color: #949494;
  color: #fff;
  padding: 5px;
  text-decoration: none;
  font-weight: bold;
  border-left: 5px solid #33ADFF;
  display: block;
}
#navbar li {
  border-left: 10px solid #666;
  border-bottom: 1px solid #666;
}
```

- Добавим стиль для выделения текста при наведении мышкой на пункт меню

```
#navbar a:hover {
  background-color: #666;
  border-left: 5px solid #3333FF;
}
```

Вертикальное меню

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      #navbar {
        margin: 0;
        padding: 0;
        list-style-type: none;
        width: 100px;
      }
      #navbar li {
        border-left: 10px solid #666;
        border-bottom: 1px solid #666;
      }
      #navbar a {
        background-color: #949494;
        color: #fff;
        padding: 5px;
        text-decoration: none;
        font-weight: bold;
        border-left: 5px solid #33ADFF;
        display: block;
      }
      #navbar a:hover {
        background-color: #666;
        border-left: 5px solid #3333FF;
      }
    </style>
  </head>
```



```
<body>

  <ul id="navbar">
    <li><a href="#">Главная</a></li>
    <li><a href="#">Новости</a></li>
    <li><a href="#">Контакты</a></li>
    <li><a href="#">О нас</a></li>
  </ul>

</body>
</html>
```

Выпадающее меню

- Меню, которое мы будем создавать, будет иметь основные навигационные ссылки, расположенные в горизонтальной панели навигации, и подпункты, которые будут отображаться только после наведения курсора мыши на тот пункт меню, к которому эти подпункты относятся.
- Сначала нам нужно создать HTML-структуру нашего меню. Основные навигационные ссылки мы поместим в маркированный список.
- Подпункты мы разместим в отдельном списке, вложив его в элемент ``, который содержит родительскую ссылку относительно подпунктов. Теперь мы имеем четкую структуру нашей будущей панели навигации:

```
<ul id="navbar">
  <li><a href="#">Главная</a></li>
  <li><a href="#">Новости</a></li>
  <li><a href="#">Контакты</a>
    <ul>
      <li><a href="#">Адрес</a></li>
      <li><a href="#">Телефон</a></li>
      <li><a href="#">Email</a></li>
    </ul>
  </li>
  <li><a href="#">О нас</a></li>
</ul>
```

Выпадающее меню

- Теперь приступим к написанию CSS кода. Для начала необходимо скрыть список с подпунктами с помощью объявления `display: none;`, чтобы они не отображались на веб-странице все время. Для отображения подпунктов нам нужно чтобы при наведении на элемент `` список снова был преобразован в блочный элемент:

```
#navbar ul { display: none; }  
#navbar li:hover ul { display: block; }
```

- Убираем у обоих списков отступы и маркеры, установленные по умолчанию. Элементы списка с навигационными ссылками делаем плавающими, формируя горизонтальное меню, но для элементов списка, содержащих подпункты задаем `float: none;`, чтобы они отображались друг под другом.

```
#navbar, #navbar ul {  
    margin: 0;  
    padding: 0;  
    list-style-type: none;  
}  
#navbar li { float: left; }  
#navbar ul li { float: none; }
```

Выпадающее меню

- Затем нам нужно сделать так, чтобы наше выпадающее подменю не смещало контент, расположенный под панелью навигации, вниз. Для этого мы зададим пунктам списка `position: relative;`, а списку, содержащему подпункты `position: absolute;` и добавим свойство `top` со значением `100%`, чтобы абсолютно позиционированное подменю отображалось точно под ссылкой.

```
#navbar ul {  
    display: none;  
    position: absolute;  
    top: 100%;  
}  
#navbar li {  
    float: left;  
    position: relative;  
}  
#navbar { height: 30px; }
```

- Высота для родительского списка была добавлена специально, так как браузеры не учитывают в качестве содержимого элемента плавающий контент, то без добавления высоты наш список будет проигнорирован браузером и контент, следующий за списком, будет обтекать наше меню.

Выпадающее меню

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Название документа</title>
    <style>
      #navbar ul{
        display: none;
        background-color: #f90;
        position: absolute;
        top: 100%;
      }
      #navbar li:hover ul { display: block; }
      #navbar, #navbar ul{
        margin: 0;
        padding: 0;
        list-style-type: none;
      }
      #navbar {
        height: 30px;
        background-color: #666;
        padding-left: 25px;
        min-width: 470px;
      }
      #navbar li {
        float: left;
        position: relative;
        height: 100%;
      }
```

```
#navbar li a {
  display: block;
  padding: 6px;
  width: 100px;
  color: #fff;
  text-decoration: none;
  text-align: center;
}
#navbar ul li { float: none; }
#navbar li:hover { background-color: #f90; }
#navbar ul li:hover { background-color: #666; }
</style>
</head>
<body>
  <ul id="navbar">
    <li><a href="#">Главная</a></li>
    <li><a href="#">Новости</a></li>
    <li><a href="#">Контакты</a>
      <ul>
        <li><a href="#">Адрес</a></li>
        <li><a href="#">Телефон</a></li>
        <li><a href="#">Email</a></li>
      </ul>
    </li>
    <li><a href="#">О нас</a></li>
  </ul>
  <p>Абзац с текстом был добавлен для демонстрации
того, что при открытии подпунктов меню они будут
скрывать часть контента, а не сдвигать его.</p>
</body>
```

Главная

Новости

Контакты

О нас

Адрес

Телефон

Email

Абзац с текстом был добавлен для демонстрации того, что при открытии подпунктов меню они будут скрывать часть контента, а не сдвигать его.

Спасибо за внимание.