

## Гиперссылки и навигация

### *HTML гиперссылки*

Гиперссылки являются основой HTML и всемирной паутины. Они представляют собой текст (или картинку), нажав на который, пользователь будет перемещен на другой документ или другое место в данном документе. По умолчанию гиперссылки выделяются на страницах подчеркнутым шрифтом синего цвета. В HTML существует 2 вида гиперссылок:

- Внешние гиперссылки перемещают пользователя нажавшего на них на другой HTML документ.
- Внутренние гиперссылки перемещают пользователя на предварительно созданную закладку в документе, в котором они определены.

Создание внешней гиперссылки:

`<a href="адрес">Текст ссылки </a>`

адрес указывает адрес документа, на который будет перемещен пользователь после нажатия на ссылку.

Для создания внутренней ссылки нужно описать закладку в документе, например укажем bookmark - имя закладки (может быть произвольным):

`<!-- Создание гиперссылки на закладку -->`

`<a href="#bookmark">Текст ссылки </a>`

`<!-- Создание закладки -->`

`<a id="bookmark">Текст закладки. </a>`

Создадим внешние ссылки ведущие на главную страницу сайта.

Пример

`<a href="https://itstep.org/ua/"> itstep.org </a>`

`<a href="https://itstep.org/ua/">`

`</a>`

Теперь создадим внутренние ссылки на закладки:

`<a href='#chapter3'>Перейти на главу 3</a>`

`<a href='#chapter5'>Перейти на главу 5</a>`

Обратите внимание: закладка в отличие от ссылки никак не выделяется на странице.

Атрибут target указывает как будет открываться документ, на который указывает ссылка. Данный атрибут может иметь следующие значения:

- \_top - документ откроется в текущем окне браузера;
- \_blank - документ откроется в новом окне браузера;
- \_self - документ откроется в текущем фрейме;
- \_parent - документ откроется в родительском фрейме.

Оформление ссылки можно изменять с помощью свойств CSS. Это псевдо классы ссылок: active, hover, link, visited. Также можно применять ряд других свойств, например,

убрать подчеркивание - **text-decoration:none**, или преобразовать элемент в блочный, используя свойство **display**.

### *Пример: раскрывающееся меню с помощью CSS*

Спецификаторы и селекторы CSS позволяют легко создать выпадающее меню. С помощью CSS3 можно создать дизайн страниц сайта, без использования фоновых изображений и Javascript, как это было раньше. Рассмотрим пошаговый процесс создания собственного раскрывающегося меню CSS. Результат должен быть следующим:



В меню, которое мы будем создавать, есть две подкатегории, которые появляются после активации родительской ссылки путем наведения курсора. Первая серия подзаголовков появляется под основной навигационной панелью, затем вторая серия ссылок отображается горизонтально с первого выпадающего списка. Структура HTML для основного меню:

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Tutorials</a></li>
    <li><a href="#">Articles</a></li>
    <li><a href="#">Inspiration</a></li>
  </ul>
</nav>
```

Сначала нужно создать структуру HTML для нашего меню CSS. Мы будем использовать теги HTML5 для размещения меню навигации в **<nav>** элементе, а затем добавим основные навигационные ссылки в простой неупорядоченный список.

```
<nav>
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Tutorials</a>
      <ul>
        <li><a href="#">Photoshop</a></li>
        <li><a href="#">Illustrator</a></li>
        <li><a href="#">Web Design</a>
          <ul>
            <li><a href="#">HTML</a></li>
            <li><a href="#">CSS</a></li>
          </ul>
        </li>
      </ul>
    </li>
  </ul>
```

```

        </li>
      </ul>
    </li>
    <li><a href="#">Articles</a>
      <ul>
        <li><a href="#">Web Design</a></li>
        <li><a href="#">User Experience</a></li>
      </ul>
    </li>
    <li><a href="#">Inspiration</a></li>
  </ul>
</nav>

```

Первые подменю нужно добавить в разделе «Tutorials» и «Articles», каждый из которых является неупорядоченным списком. Второе подменю вложено в подпункт «Web Design» первого подменю. Эти ссылки помещаются в другой неупорядоченный список и вставляются в раздел «Web Design».

- [Home](#)
- [Tutorials](#)
  - [Photoshop](#)
  - [Illustrator](#)
  - [Web Design](#)
    - [HTML](#)
    - [CSS](#)
- [Articles](#)
  - [Web Design](#)
  - [User Experience](#)
- [Inspiration](#)

На данном шаге мы расположили ссылки в зависимости от родительских элементов, теперь приступим к оформлению. Стиль CSS для вложенных элементов:

```

nav ul ul {
  display: none;
}

nav ul li:hover > ul {
  display: block;
}

```

Сначала подготовим базовую функциональность выпадающего списка. С помощью спецификаторов CSS и расширенных селекторов мы можем настраивать отдельные элементы, глубоко расположенные в структуре HTML, без необходимости в дополнительных идентификаторах или классах. Сначала скроем подменю, расположенные внутри указав свойство **display:none**. Чтобы снова открыть эти меню, они должны быть преобразованы обратно в блочные элементы, при наведении на соответствующий **LI**. Дочерний селектор (>) позволяет выбрать только дочерний элемент **UL**, а не все подменю.



```
nav ul {
  background: #efefef;
  background: linear-gradient(top, #efefef 0%, #bbbbbb 100%);
  background: -moz-linear-gradient(top, #efefef 0%, #bbbbbb 100%);
  background: -webkit-linear-gradient(top, #efefef 0%, #bbbbbb 100%);
  box-shadow: 0px 0px 9px rgba(0,0,0,0.15);
  padding: 0 20px;
  border-radius: 10px;
  list-style: none;
  position: relative;
  display: inline-table;
}
nav ul:after {
  content: "";
  clear: both;
  display: block;
}
```

Затем мы можем приступить к стилизации главного меню навигации с помощью свойств CSS3, таких как градиенты, тени и радиус скругления. Добавление **position:relative** позволит нам полностью позиционировать подменю в соответствии с расположением основной навигационной панели, а затем **display:inline-table** уплотнит ширину меню. Правило стиля **clear** очищает обтекание, используемые в последующих элементах списка, без необходимости использования **overflow:hidden**, которые скрывали бы подменю и не отображали их.



```
nav ul li {
  float: left;
}
```

```

nav ul li:hover {
    background: #4b545f;
    background: linear-gradient(top, #4f5964 0%, #5f6975 40%);
    background: -moz-linear-gradient(top, #4f5964 0%, #5f6975 40%);
    background: -webkit-linear-gradient(top, #4f5964 0%, #5f6975 40%);
}

nav ul li:hover a {
    color: #fff;
}

nav ul li a {
    display: block;
    padding: 25px 40px;
    color: #757575;
    text-decoration: none;
}

```

Отдельные пункты меню оформляются с помощью правил CSS, добавленных к элементу `<a>` вложенному в `<li>`. В браузере это приведет к изменению оформления ссылки на фоне синего градиента и белого текста.



```

nav ul ul {
    background: #5f6975;
    border-radius: 0px;
    padding: 0;
    position: absolute;
    top: 100%;
}

nav ul ul li {
    float: none;
    border-top: 1px solid #6b727c;
    border-bottom: 1px solid #575f6a;
    position: relative;
}

nav ul ul li a {
    padding: 15px 40px;
    color: #fff;
}

nav ul ul li a:hover {
    background: #4b545f;
}

```

Основная панель навигации теперь стилизована, но в подменю все еще нужна определенная работа. В настоящее время они наследуют стили от своих родительских элементов, поэтому изменение фона и удаление радиуса и отступов границы изменяет их внешний вид. Чтобы убедиться, что они отображаются под основным меню, они позиционированы абсолютно на 100% от верхнего края UL (т.е. снизу).

Для каждого **LI** в подменю не нужны обтекание, вместо этого они перечислены вертикально с тонкими границами, разделяющими каждый пункт. При наведении получаем эффект затемнения фона.



```
nav ul ul ul {  
    position: absolute;  
    left: 100%;  
    top: 0;  
}
```

Заключительный шаг - поместить под-подменю соответственно. Эти меню уже наследуют все оформление, поэтому им нужно указать только расположение, позиционирование - абсолютное, размер отступа слева: 100%, а сверху 0, относительно положения родителя **LI**.

Выпадающее меню завершено.

