# Лабораторная работа №1

**CSS-Меню: горизонтальное, вертикальное, выпадающее**

1. **Цель работы:**
   1. Научиться работать с идентификаторами
   2. Научиться позиционировать элементы на странице.
   3. Научиться создавать различные виды меню.
2. **Приборы и оборудование:**
   1. Методические указания.
   2. ПЭВМ типа IBM PC/XT.
   3. Текстовый редактор Блокнот/ Sublime Text.
3. **Порядок выполнения работы:**
   1. Изучить основные теоретические сведения.
   2. Запустить текстовый редактор Блокнот/ Sublime Text.
   3. Разработать главную страницу для музыкальной группы (по вашему усмотрению). Создать html страницу, содержащую следующие элементы:

- шапка сайта, содержащая название группы

- навигационное меню (вертикальное или горизонтальное)

- область контента, содержащая краткую информацию о группе

- подвал сайта, содержащий информацию о разработчиках, ссылки обратной связи в социальных сетях и ссылку на возврат на начало страницы.

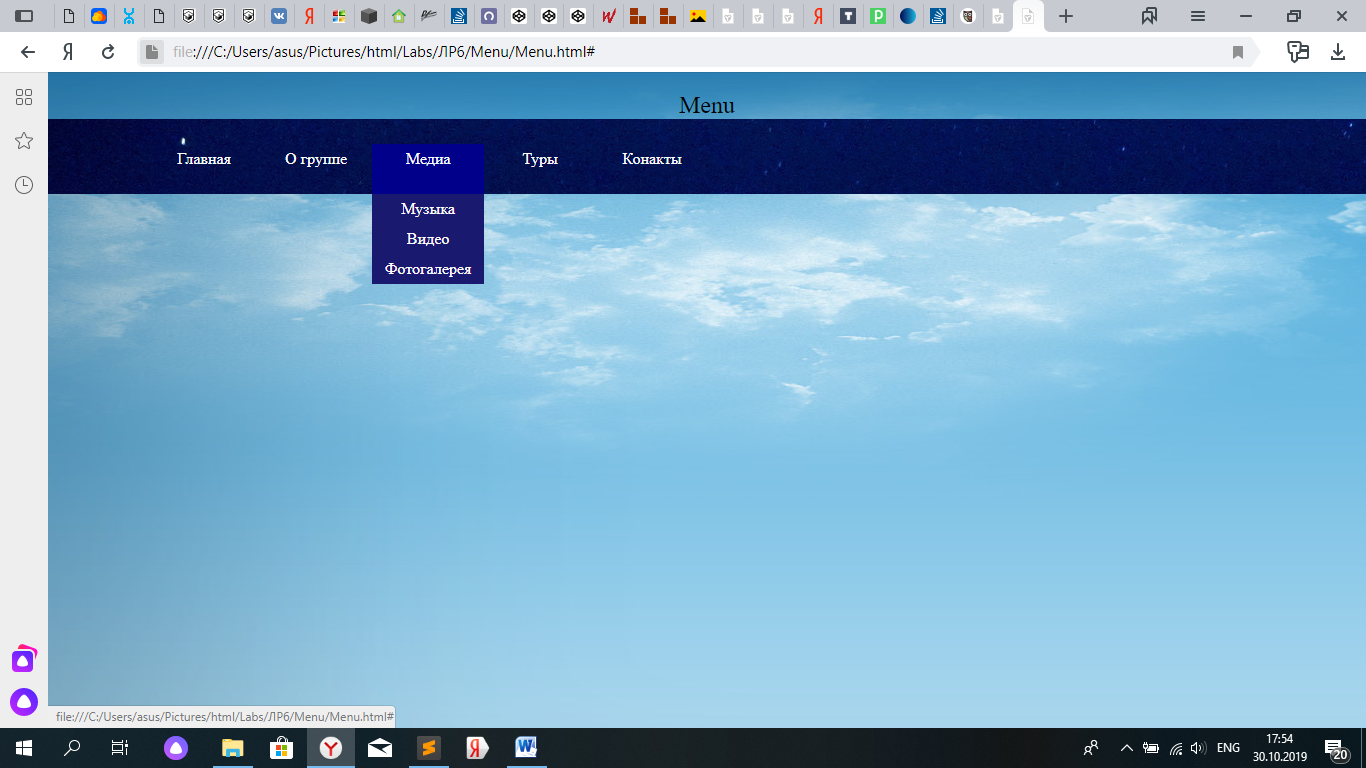


Рисунок №1 Примерное содержание меню

* 1. Сформулируйте выводы по проделанной работе. Оформите отчёт.

1. **Контрольные вопросы:**
   1. Что представляет из себя селектор идентификатора CSS?
   2. Как можно подключить идентификатор CSS?
   3. Какие типы селекторов вы знаете?
   4. Для чего необходимо позиционирование элементов? Какие виды позиционирования вы знаете?

Приложение А

(теоретические сведения)

**Селектор идентификатора CSS**

CSS селектор id позволяет применить стиль к определённому элементу на странице, в котором указан необходимый идентификатор. Определение селектора id в CSS коде начинается с символа решётки "#", за которым следует имя идентификатора:

|  |  |
| --- | --- |
| 1 | #myid { color: blue; } |

Чтобы использовать селектор id, нужно указать, к какому элементу на странице вы хотите его применить, для этого надо добавить атрибут id в открывающем теге и указать в качестве значения имя нужного идентификатора. Ставить решётку перед именем идентификатора в HTML коде (в значении атрибута id) не нужно.

**Примечание:** идентификаторы не должны повторяться в рамках одного документа, т.е. определенный id может быть использован на странице только один раз с тегом, для которого он предназначен. Если вы хотите применять стиль более чем к одну элементу, следует использовать селектор class.

Селектор id имеет некоторые особенности:

* В имени идентификатора разрешается использовать только буквы, числа, дефис и знак подчеркивания
* Имя идентификатора всегда должно начинаться с буквы
* Имена идентификаторов чувствительны к регистру символов, например: **#Menu** и **#menu** это два разных идентификатора

**Создание вертикального меню**

1. 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 <li> нашего списка будет содержать по одной ссылке.
2. Наша следующая задача заключается в сбросе стилей списка, установленных по умолчанию. Нам нужно убрать внешние и внутренние отступы у самого списка и маркеры у пунктов списка. Затем зададим нужную ширину
3. Теперь пришло время стилизовать сами ссылки. 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.
4. 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.
5. При наведении курсора мыши на пункт меню его внешний вид может изменяться, привлекая к себе внимание пользователя. Создать такой эффект можно с помощью псевдо-класса :hover.

**Создание горизонтального меню**

В предыдущем примере мы рассмотрели вертикальную панель навигации, которую чаще всего можно встретить на сайтах слева или справа от области с основным контентом. Однако меню с навигационными ссылками также часто располагается и по горизонтали в верхней части веб-страницы.

1. Горизонтальное меню можно создать путем стилизации обычного списка. Свойству display для элементов <li> нужно присвоить значение inline, чтобы пункты списка располагались друг за другом.
2. Для размещения пунктов меню по горизонтали, сначала создадим маркированный список с ссылками
3. Напишем для нашего списка пару правил, сбрасывающих стиль используемый для списков по умолчанию, и переопределим пункты списка с блочных на строчные.
4. Теперь нам осталось лишь определить стилевое оформление для нашего горизонтального меню.

**Создание выпадающего списка меню**

1. Подпункты мы разместим в отдельном списке, вложив его в элемент <li>, который содержит родительскую ссылку относительно подпунктов.
2. 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 <li> список снова был преобразован в блочный элемент
3. 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.
4. 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.

С помощью позиционирования можно очень точно и гибко управлять расположением элементов. Позиционирование применяют для создания сложных «многослойных» интерфейсов, таких как всплывающие окна или галереи, а также для вёрстки мелких декоративных элементов.

Главным свойством для работы с позиционированием является position, которое переключает режимы позиционирования элемента. Ещё четыре свойства: top, right, bottom, left, управляют расположением элемента. И последнее свойство z-index управляет порядком слоёв.

Типичный веб-сайт состоит из шапки, главного меню, блока с основным содержанием, боковых колонок, подвала. Эти блоки могут быть расположены друг под другом, в несколько колонок или ещё сложнее. Такое взаимное расположение основных блоков сайта и называют «сеткой» или «раскладкой».

Сетка может быть фиксированной или тянущейся, когда ширина блоков изменяется в зависимости от ширины браузера. Чтобы быстро и легко создавать сетки, нужно хорошо разбираться в блочной модели и позиционировании.

**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.

### Синтаксис

float: left | right | none | inherit

### 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 задается.

inherit

Наследует значение родителя.

Позиционирование позволит Вам разместить тот, или иной элемент в том месте, где это Вам необходимо, цель этой статьи заключается в том, чтобы понять по каким правилам это происходит, какие при этом необходимо использовать CSS свойства и для чего.

## Типы позиционирования элементов

Основное свойство CSS, которое позволяет управлять позиционированием элементов на странице это свойство **[position](https://basicweb.ru/css/css_pr_position.php" \o "CSS свойство position)**, оно сообщает браузеру, какой тип позиционирования используется для элемента (*статический* - **static**, *относительный* - **relative**, *абсолютный* – **absolute**, или *фиксированный* - **fixed**).

Для полного понимания как происходит позиционирование элементов на любой странице, Вам необходимо детально изучить все виды позиционирования. Эта статья учебника предоставит Вам такую возможность, сейчас мы с Вами отдельно поговорим о каждом виде позиционирования и разберем, как и относительного чего происходит смещение элементов в документе.

### Абсолютное позиционирование

Совместно со свойством **[position](https://basicweb.ru/css/css_pr_position.php" \o "CSS свойство position)** используются CSS свойства, которые управляют смещением позиционированного элемента:

* [**top**](https://basicweb.ru/css/css_pr_top.php) (смещение позиционированного элемента от верхнего края).
* [**right**](https://basicweb.ru/css/css_pr_right.php) (смещение позиционированного элемента от правого края).
* [**bottom**](https://basicweb.ru/css/css_pr_bottom.php) (смещение позиционированного элемента от нижнего края).
* [**left**](https://basicweb.ru/css/css_pr_left.php) (смещение позиционированного элемента от левого края).

В качестве значений, которые определяют смещение элемента, допускается использовать **физические единицы** (например, *пункты*), но чаще используют **визуальные единицы** – *пиксели*, **процентные значения** и значения *em*. Значения могут быть как *положительные*, так и *отрицательные*, как и люди, только значения.

При использовании абсолютного позиционирования (**[position](https://basicweb.ru/css/css_pr_position.php" \o "CSS свойство position)**: **absolute**) элемент сдвигается (позиционируется) **относительно заданного края его предка**, при этом предок **должен иметь**значение **[position](https://basicweb.ru/css/css_pr_position.php" \o "CSS свойство position)** отличное от, установленного по умолчанию - **static**, иначе отсчёт (смещение) будет вестись **относительно, указанного края окна браузера**.

### Относительное позиционирование

Следующий тип позиционирования, который мы рассмотрим это *относительное позиционирование*. Элементы, для которых задано *относительное позиционирование* (**[position](https://basicweb.ru/css/css_pr_position.php" \o "CSS свойство position)**: **relative**) смещаются (размещаются) **относительно положения в потоке документа**, или другими словами **относительно его текущей позиции**.

### Фиксированное позиционирование

*Третий* тип позиционирования, который мы рассмотрим это *фиксированное позиционирование*. При *фиксированном позиционировании* элемент сдвигается **относительно заданного края окна браузера**. Отличительная особенность этого позиционирования заключается в том, что при прокрутке страницы **элемент остается на одном месте**, то есть, грубо говоря, он прокручивается вместе со страницей (элемент зафиксирован).