

Лабораторная работа № 10

Адаптивная веб-дизайн. Flexbox-верстка

Цель работы: изучить принципы адаптивного веб-дизайн и использования медиазапросов, получить навыки применения flexbox-верстки.

Теоретические сведения для выполнения работы

Понятие адаптивный веб-дизайн. Медиазапросы

Адаптивная веб-дизайн позволяет изменять структуру веб-страницы на основе ширины окна браузера на различных устройствах (планшеты, смартфоны). Для его осуществления используются гибкие сетки, гибкие изображения и медиазапросы CSS, предназначенные для создания различных стилей для экранов с разным решением.

Существует несколько методов по созданию адаптивного веб-дизайна:

1. **Стратегия Mobile First** заключается в разработке мобильной версии до создания настольной версии.

2. **Правило @media** с помощью которого стили адаптируются под разные области просмотра.

3. **Использование резиновых макетов** допускает масштабирование контейнеров в зависимости от ширины области просмотра.

Для отображения содержимого страницы на экране смартфона можно использовать следующий метатег:

```
<meta name="viewport" content="width=device-width">
```

Браузеры не уменьшают масштаб, а настраивают ширину экрана на текущее разрешение по горизонтали экрана смартфона.

Медиазапросы назначают страницам стили на основе размера окна браузера. С помощью них можно создавать пользовательские стили для браузеров смартфонов, планшетов и компьютеров и производить настройку отображения сайта на экране каждого типа устройств. Медиазапрос с помощью правила **@media** выглядит следующим образом:

```
@media (min-width: 480px) {  
  div > h1 {  
    font-size: 2.25rem;  
  }  
}
```

В данном случае браузер проверяет, выполняется ли условие **min-width: 480px** и если размер экрана больше или 560 пикселей, то применяются заданные стили.

Также можно использовать атрибут **media** в HTML-коде следующим образом:

```
<link href="css/small.css" rel="stylesheet" media="(max-width: 480px)">
```

В этом случае стиль small.css будет применяться, если разрешение не более 480px.

Различные ширины экрана, которые указываются с помощью медиазапросов называются точками останова.

Правило **@import** также можно использовать для создания медиазапросов. Оно должно помещаться в начало таблицы стилей и подключается следующим образом:

```
@import url(css/base.css); /* без медиазапроса */
@import url(css/medium.css) (min-width:560px) and (max-width:760);
@import url(css/small.css) (max-width: 480px);
```

Основным компонентом адаптивного дизайна являются гибкие сетки, которые являются частью резинового дизайна. Применение гибких сеток заключается в установке процентного значения ширины вместо абсолютного в пикселах.

<pre><div class="columns"> <div class="one-third"> ...контент... </div> <div class="two-thirds"> ...контент... </div> </div></pre>	<pre>.columns { width: auto; max-width: 1200px; } .one-third { float: left; width: 33%; } .two-thirds { float: left; width: 67%; }</pre>
--	--

Различные ширины экрана, которые указываются с помощью медиазапросов называются точками останова. Стандартными принято считать следующие разрешения: для мобильных устройств 480px, для планшетных компьютеров 768 px, для нетбуков 1024 px, для ПК 1280 px и больше.

Flexbox-верстка

Flexbox-верстка — разметка макетов веб-страниц, которая позволяет автоматически настраивать ширину элементов, находящихся внутри flex-контейнера. Для применения данного способа разметки следует учитывать разделение применяемых свойств на свойства flex-контейнера и свойства flex-элемента.

С помощью ***display: flex;*** осуществляется преобразование HTML-элемента в flex-контейнер, а элементы находящийся внутри него в flex-элементы. По умолчанию flex-элементы помещаются друг за другом в одной строке. Свойство ***flex-flow*** позволяет выбрать направление отображения элементов в контейнере, а также указать их перенос на следующую строку. Первое значение определяет направление, а второе перенос на следующую строку.

Свойство ***flex-flow*** является сокращенной записью и содержит значения свойств ***flex-direction*** и ***flex-wrap***. Flex-элементы могут размещаться в строке

(значение **row**) или в колонке (значение **column**). Перенос элементов осуществляется с помощью значения **wrap**.

Свойство *justify-content* определяет способ выключки flex-элементов в строке. Для выравнивания левому краю используется значения **flex-start**, а по правому краю — **flex-end**. Для равномерного распределения элементов создавая пространство между ними необходимо значение **space-between**, а **space-around** добавляет поля по левому и правому краям крайних элементов.

Свойство *align-items* определяет, как flex-элементы различной высоты будут выровнены по высоте строки в flex-контейнере. Для выравнивания верхнему краю используется значения **flex-start**, а по нижнему краю — **flex-end**. Значение **stretch** позволяет растянуть каждый элемент по высоте контейнера, делая их высоты одинаковыми. Свойство *align-content* определяет как будут размещены flex-элементы, занимающие несколько строк.

Для flex-элементов основным свойством является *flex*, которое обеспечивает их гибкость и управляет шириной, что позволяет создавать «гибкие» колонки или изменять их ширину в соответствии с размером контейнера, даже если размер неизвестен или меняется динамически. Первое значение свойства flex — число параметра *flex-grow*, которое указывает на относительную ширину flex-элемента, которая определяет во сколько размеры элементов отличаются между собой. Второе значение — число свойства *flex-shrink*, которое определяет насколько flex-элемент может быть сжат, если суммарная ширина элементов больше ширины контейнера. Последнее значение — свойство *flex-basis*, которое определяет базовую ширину flex-элемента. Пример кода с возможным свойствами представлен в таблице.

Структура страницы	Значения flex-контейнера	Значения flex-элемента
<pre><div class="container"> <div class="div1"></div> <div></div> </div></pre>	<pre>.container { display: flex; flex-flow: row wrap; align-content: space-between; align-items: flex-start; justify-content: center; }</pre>	<pre>.div1 { order: 1; align-self: flex-end; flex: 1 1 300px; }</pre>

Задания к лабораторной работе № 10

Задание 1 Преобразуйте макет представленный на рис. 10.1, созданный в задании 1 лаб.раб. № 9 с использованием свойств flexbox-верстки, сделав предварительно копию. Для макета задать возможность переноса элементов на новую строку. Добавьте дополнительно 2 ячейки, заполнив информацией на тему «Flexbox-верстка»

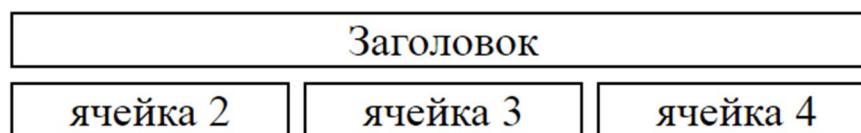


Рис. 10.1

Задание 2 Сделайте в новом документе горизонтальное меню таким образом, чтобы при изменении экрана до 560px оно преобразовывалось в вертикальное выпадающее.

Задание 3 Создайте новый HTML-документ под название video.html по макету согласно рис. 10.2 следующим образом:

3.1 Добавить два аудиофайла.

3.2 Добавить три видеофайла.

3.3 При изменении размера экрана меню должно становиться выпадающим, а под ним располагаться аудио и видео контент.

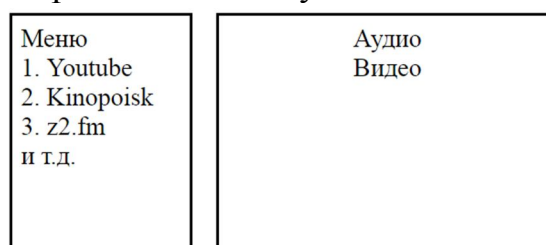


Рис. 10.2

Задание 4 В документе к заданию 1 добавьте 4 фотографии с надписью и продемонстрируйте подключение разными способами медиазапросов, которые будут изменять поведение фотографий и размер шрифта надписи в зависимости от размера экрана для 480px, 768px, 1280px.

Примечание: медиазапросы <https://html5book.ru/css3-mediazaprosy/>
 пример адаптивного меню <https://sitespectr.ru/adaptivnoe-menu/>

Контрольные вопросы

1. Что представляет из себя фиксированная верстка веб-страницы?
2. Что такое резиновый макет веб-страницы?
3. Дайте понятие «адаптивный дизайн»
4. Как создать адаптивные веб-сайты?
5. Что такое медиазапросы?
6. Как подключить медиазапросы?
7. Для чего используется правило `@import`?
8. Для чего используется правило `@media`?
9. Для чего предназначено свойство `flex`?
10. Что означает `@media (min-width: 560px)`?
11. Что означает `@media (max-width: 960px)`?
12. Что означает `flex: 1 2 200px;`?
13. К какому свойству относится и для чего предназначено значение `wrap`?
14. Для чего используется свойство `flex-flow`?

15. Для чего используется свойство *flex-direction*?
16. Для чего предназначены свойство *align-items* и свойство *align-content*?
17. Какие значения имеет свойство *align-self*?
18. Какие свойства имеет flex-контейнер?
19. Какие свойства имеет flex-элементы?
20. Для чего предназначено свойство *justify-content*?
21. Как вставить видео? Как вставить аудио?
22. Для чего используется свойство *flex-wrap*?
23. Чем отличается *align-items* от свойства *align-self*?
24. Как создать адаптивное меню?