

Лабораторная работа № 10

АДАПТИВНЫЙ ВЕБ-ДИЗАЙН. FLEXBOX-ВЕРСТКА

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

Теоретические сведения для выполнения работы

Понятие адаптивный веб-дизайн. Медиазапросы

Адаптивный 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. Метод **Mobile First**, который 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, которое 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. В мобильной версии часто используется значок в виде трех горизонтальных линий, который называют гамбургером. Для вставки символ значка тройного равно в стандарте Юникод предусмотрено для html `≡` и css `\2261`.

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;}  
}
```

Рис. 10.1 Пример синтаксиса медиазапроса

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 `<link href="css/small.css" rel="stylesheet" media="(max-width: 480px)">`. В этом случае стиль `small.css` будет применяться, если разрешение не более 480px.

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);
```

Рис. 10.2 Пример использования правила @import

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

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 ши-рину flex-элемента. Пример кода с возможным свойствами представлен в таблице 10.

Таблица 10.1

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

Задания к лабораторной работе № 10

Задание 1 Создайте макет представленный на рис. 10.3 с использованием свойств flexbox-верстки, сделав предварительно копию. В ячейку под номером 1 добавить три видео файла, в ячейке под номером два должна быть информация на тему «Flexbox-верстка», в третьей ячейке должно быть три аудиофайла.



Рис. 10.3 Макет для задания 1

Задание 2 В этом же документе создайте горизонтальное меню, которое в мобильной версии преобразуется в меню гамбургер. Вид мобильной версии представлен на рис. 10.4

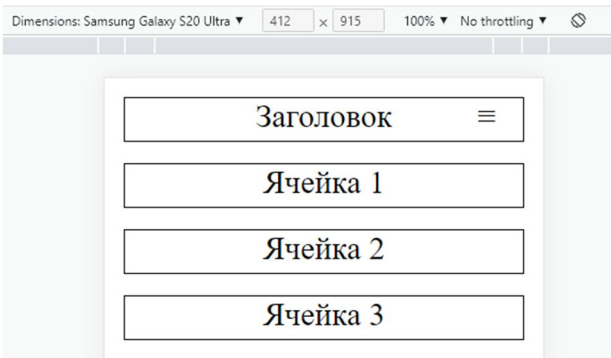


Рис. 10.4 Вид мобильной версии для задания 2

Самостоятельно подобрать ширину экрана и способ подключения для медиазапроса.

Задание 3 В документе к заданию 1 добавьте фотогалерею из 8 фотографий с надписью и продемонстрируйте подключение разными способами медиазапросов, которые будут изменять цвет шрифта надписи в зависимости от размера экрана: 480px — красный, внутреннее подключение, 768px — синий, использовать правило `@import`, 1280px — зеленый, через тег `link`. Также следует для размера экранов меньше 1280px установить размер области занимаемой одной фотографией равной ширине экрана, расположение фотографий сделать вертикальным.

Контрольные вопросы

1. Что такое фиксированная верстка веб-страницы?
2. Что такое резиновый макет веб-страницы?
3. Дайте понятие «адаптивный дизайн»
4. Что такое медиазапросы? Как подключить медиазапросы?
5. Для чего используется правило `@import`?
6. Для чего используется правило `@media`?
7. Для чего предназначено свойство *flex*?
8. Что означает `@media (min-width: 560px)`?
9. Что означает `@media (max-width: 960px)`?
10. Что означает *flex: 1 2 200px*;
11. Для чего предназначено значение *wrap*?
12. Для чего используется свойство *flex-flow*?
13. Для чего используется свойство *flex-direction*?
14. Для чего свойство *align-items* и свойство *align-content*?
15. Какие значения имеет свойство *align-self*?
16. Какие свойства имеет flex-контейнер?
17. Какие свойства имеет flex-элементы?
18. Для чего предназначено свойство *justify-content*?
19. Как вставить видео? Как вставить аудио?
20. Чем отличается *align-items* от свойства *align-self*?
21. Как создать адаптивное меню?
22. Создайте документ, в котором цвет абзаца изменяется на красный при изменении ширины до 768px
23. Что такое специфичность селектора? Какие значения она может иметь?