# 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** c 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 сматрфона.

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: 480рх) {

div > hl { font-size: 2.25rem;

}

}

В данном случае браузер проверяет, выполняется ли условие ***min-width: 480рх*** и если размер экрана больше или 560 пикселов, 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 случает 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.

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

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 flexконтейнера. Для применения данного способа разметки следует учитывать разделение применяемых свойств на свойства flex-контейнера и свойства flex-элемента.

С помощью ***display: flex;*** осуществляется преобразование HTMLэлемента в flex-контейнер, а элементы находящийся внутри него в flexэлементы. 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 запись 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 — **flexend**. 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 являет *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** |
| <div class="container">  <div class="div1"></div>  <div></div>  </div> | .container { display: flex; flex-flow: row wrap;  align-content: spacebetween;  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** 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»

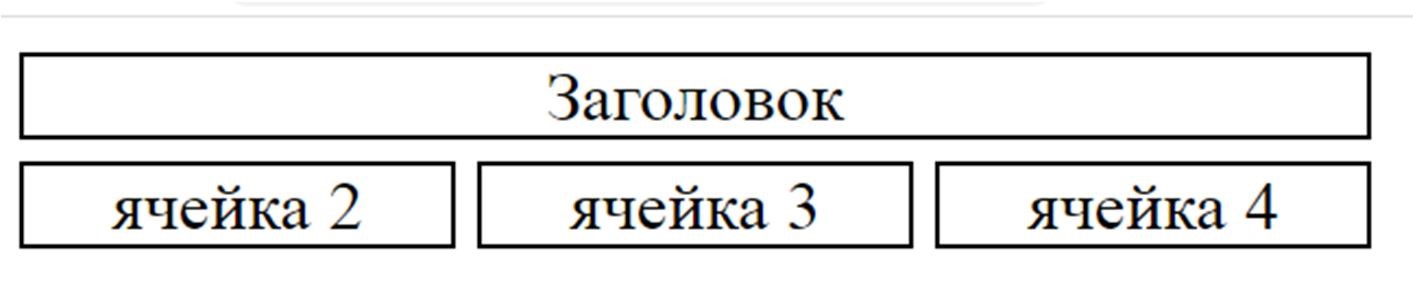


Рис. 10.1

**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.

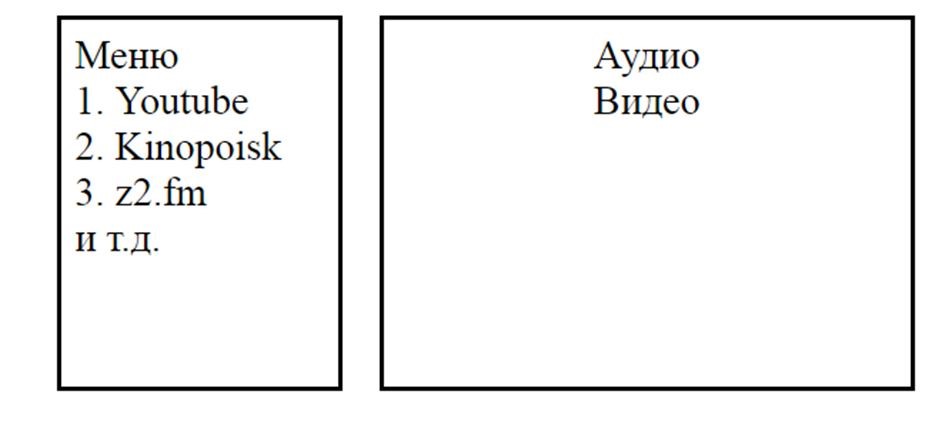


Рис. 10.2

**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?

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

1. Что такое резиновый макет веб-страницы?

**Резиновый** (гибкий) **макет** сайта - это **макет**, который изменяется в зависимости от ширины рабочей области окна (вкладки) браузера.

3. Дайте понятие «адаптивный дизайн»

**Адаптивный** веб-**дизайн** — это **дизайн** страниц в интернете, который позволяет получить правильное отображение веб-ресурса на любых устройствах, подключенных к сети

1. Как создать адаптивные веб-сайты?
2. Что такое медиазапросы?

**Медиа** **запросы** (media queries) - **это** правила CSS, которые позволяют управлять стилями элементов в зависимости от значений технических параметров устройств.

1. Как подключить медиазапросы?

@media (min-width: 480рх) {

div > hl { font-size: 2.25rem; }

1. Для чего используется правило @import?

Правило @**import** позволяет импортировать внешние таблицы стилей не только в HTML-страницу, но и в другой CSS.

1. Для чего используется правило @media?

CSS **правило** @**media** используется для изменения CSS стилей на основании определенных характеристик устройства, связанных с отображением контента, таких как ширина, высота, ориентация, разрешение экрана

1. Для чего предназначено свойство *flex*?

С помощью ***display: flex;*** осуществляется преобразование HTMLэлемента в flex-контейнер, а элементы находящийся внутри него в flexэлементы. По умолчанию flex-элементы помещаются друг за другом в одной строке.

1. Что означает **@media (min-width: 560рх)**?
2. Что означает **@media (max-width: 960рх)**?
3. Что означает ***flex: 1 2 200px;***?
4. К какому свойству относится и для чего предназначено значение *wrap*?

*flex-wrap*

Flex-элементы переносятся, располагаясь в несколько горизонтальных рядов (если не помещаются в один ряд) в направлении слева направо (в rtl справа налево).

1. Для чего используется свойство *flex-flow*?

Свойство позволяет определить направления главной и поперечной осей, а также возможность переноса flex-элементов при необходимости на несколько строк. Представляет собой сокращённую запись свойств flex-direction и flex-wrap.

1. Для чего используется свойство *flex-direction*?

Свойство относится к flex-контейнеру. Управляет направлением главной оси, вдоль которой укладываются flex-элементы, в соответствии с текущим режимом записи. Не наследуется.

1. Для чего предназначены свойство *align-items* и свойство *aligncontent*?

Свойство выравнивает flex-элементы, в том числе и анонимные flex-элементы по поперечной оси. Не наследуется.

Свойство выравнивает строки в flex-контейнере при наличии дополнительного пространства на поперечной оси, аналогично выравниванию отдельных элементов на главной оси с помощью свойства justify-content. Свойство не влияет на однострочный flex-контейнер. Не наследуется

Какие значения имеет свойство *align-self*?

|  |  |
| --- | --- |
| auto | Значение по умолчанию. Flex-элемент использует выравнивание, указанное в свойстве align-items flex-контейнера. |
| flex-start | Верхний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей margin и рамок border элемента), проходящей через начало поперечной оси. |
| flex-end | Нижний край flex-элемента помещается вплотную с flex-линией (или на расстоянии, с учетом заданных полей margin и рамок border элемента), проходящей через конец поперечной оси. |
| center | Поля flex-элемента центрируется по поперечной оси в пределах flex-линии. |
| baseline | Flex-элемент выравнивается по базовой линии. |

1. Какие свойства имеет flex-контейнер?

После установки данных значений свойства каждый дочерний элемент автоматически становится flex-элементом, выстраиваясь в один ряд (вдоль главной оси). При этом блочные и строчные дочерние элементы ведут себя одинаково, т.е. ширина блоков равна ширине их содержимого с учетом внутренних полей и рамок элемента.

1. Какие свойства имеет flex-элементы?

Flex-элементы — блоки, представляющие содержимое flex-контейнера в потоке. Flex-контейнер устанавливает новый контекст форматирования для своего содержимого, который обуславливает следующие особенности:

* Для flex-элементов блокируется их значение свойства display. Значение display: inline-block; и display: table-cell; вычисляется в display: block;.
* Пустое пространство между элементами исчезает: оно не становится своим собственным flex-элементом, даже если межэлементный текст обернут в анонимный flex-элемент. Для содержимого анонимного flex-элемента невозможно задать собственные стили, но оно будет наследовать их (например, параметры шрифта) от flex-контейнера.
* Абсолютно позиционированный flex-элемент не участвует в компоновке гибкого макета.
* Поля margin соседних flex-элементов не схлопываются.
* Процентные значения margin и padding вычисляются от внутреннего размера содержащего их блока.
* margin: auto; расширяются, поглощая дополнительное пространство в соответствующем измерении. Их можно использовать для выравнивания или раздвигания смежных flex-элементов.
* Автоматический минимальный размер flex-элементов по умолчанию является минимальным размером его содержимого, то есть min-width: auto;. Для контейнеров с прокруткой автоматический минимальный размер обычно равен нулю.

1. Для чего предназначено свойство *justify-content*?

Свойство выравнивает flex-элементы по главной оси flex-контейнера, распределяя свободное пространство, незанятое flex-элементами. Когда элемент преобразуется в flex-контейнер, flex-элементы по умолчанию сгруппированы вместе (если для них не заданы поля margin).

1. Как вставить видео? Как вставить аудио?
2. Для чего используется свойство *flex-wrap*?
3. Чем отличается *align-items* от свойства *align-self*?

Flex-элементы можно выравнивать по поперечной оси текущей строки flex-контейнера. align-items устанавливает выравнивание для всех элементов flex-контейнера, включая анонимные flex-элементы. align-self позволяет переопределить это выравнивание для отдельных flex-элементов.

1. Как создать адаптивное меню?