**Практическое занятие №16**

**Создание макета посадочной страницы с помощью адаптивной верстки**

**Цель**:

* научиться создавать макет посадочной страницы с помощью адаптивной верстки

**План:**

* Теоретические сведения
* Ход работы
* Контрольные вопросы

**Теоретические сведения**

В последнее время все больше и больше развиваются технологии и различные устройства (планшеты, смартфоны, мониторы) при помощи, которых просматриваются сайты.

По этой причине получаем огромное множество различных разрешений экранов, что в свою очередь ведет к затруднению получения информации с сайтов. Чтобы информация была хорошо читаема  и удобно просматриваема на большинстве устройств и была разработана технология адаптивного дизайна.

**Целью адаптивного дизайна** является разработка универсального дизайна веб-сайта, который бы в свою очередь позволял просматривать и взаимодействовать с ресурсом с различных устройств.

А также данная технология предполагает разработку одной версии веб-сайта для всех устройств, а не нескольких.

**Адаптивный веб-дизайн** (англ. AdaptiveWebDesign) – дизайн веб-страниц, обеспечивающий корректное отображение сайта на различных устройствах, подключённых к интернету и динамически подстраивающийся под заданные размеры окна браузера.

**Основные принципы адаптивного дизайна:**

* **адаптивный шаблон сайта**, способность шаблона подстраиваться под различные разрешения экранов устройств от монитора компьютера до смартфона;
* **адаптация и перемещение блоков контента**, способность блоков контента в зависимости от разрешения экрана устройства принимать необходимые размеры, а также способность передвигаться на другую позицию в макете;
* **адаптация изображений**, способность изображений менять размер в зависимости от разрешения экрана или загружать более адаптированное изображение с меньшим весом и размером;
* **использование гибкой сетки**, позволяет максимально быстро изменять конструкцию макета;
* **скрытие менее важных блоков**, на небольших экранах некоторые блоки могут скрываться;
* **переработка юзабилити элементов навигации**, так как на мобильных устройствах в связи с небольшим разрешением элементы навигации становятся менее кликабельными, их перерабатывают, делая удобно-используемыми;
* **упрощение ряда элементов на веб - странице**, некоторые элементы упрощаются для использования на мобильных устройствах;
* **адаптация видео-контента**, также следует учесть и адаптацию видео;
* **использование медиа-запросов** (mediaquery), позволяют создавать адаптивный макет.

Многие утверждают, что необходимо начинать разработку макета именно с версии для мобильных устройств. Но все же проще разработать макет под максимальное разрешение с продуманной сеткой и полным функционалом, а уже потом адаптировать под другие разрешения.

**Размеры макетов адаптивного дизайна**

Если следовать принципу сначала мобильные то будут такие размеры разрешений, под которые необходимо разрабатывать макет:

Смартфон – **320 x 480 пикселей**

Планшет – **768 x 1024 пикселей**

Компьютер (десктоп) – **1440 x 900 пикселей**может быть и больше зависит от задач.

Картина выглядит примерно, таким образом, но зачастую под какие-то разрешения нет необходимости создавать макет, к примеру, под **480px,** если макет не ломается на промежутке **768 - 320px.**

Создавая холст в фотошопе, нужно учитывать отступы, полосы прокрутки и все остальное. Создают модульную сетку для удобства разработки, а также более быстрого проектирования макета. Многие используют в работе фреймворки, а следовательно опираются на сетку того фреймворка на котором ведут разработку.

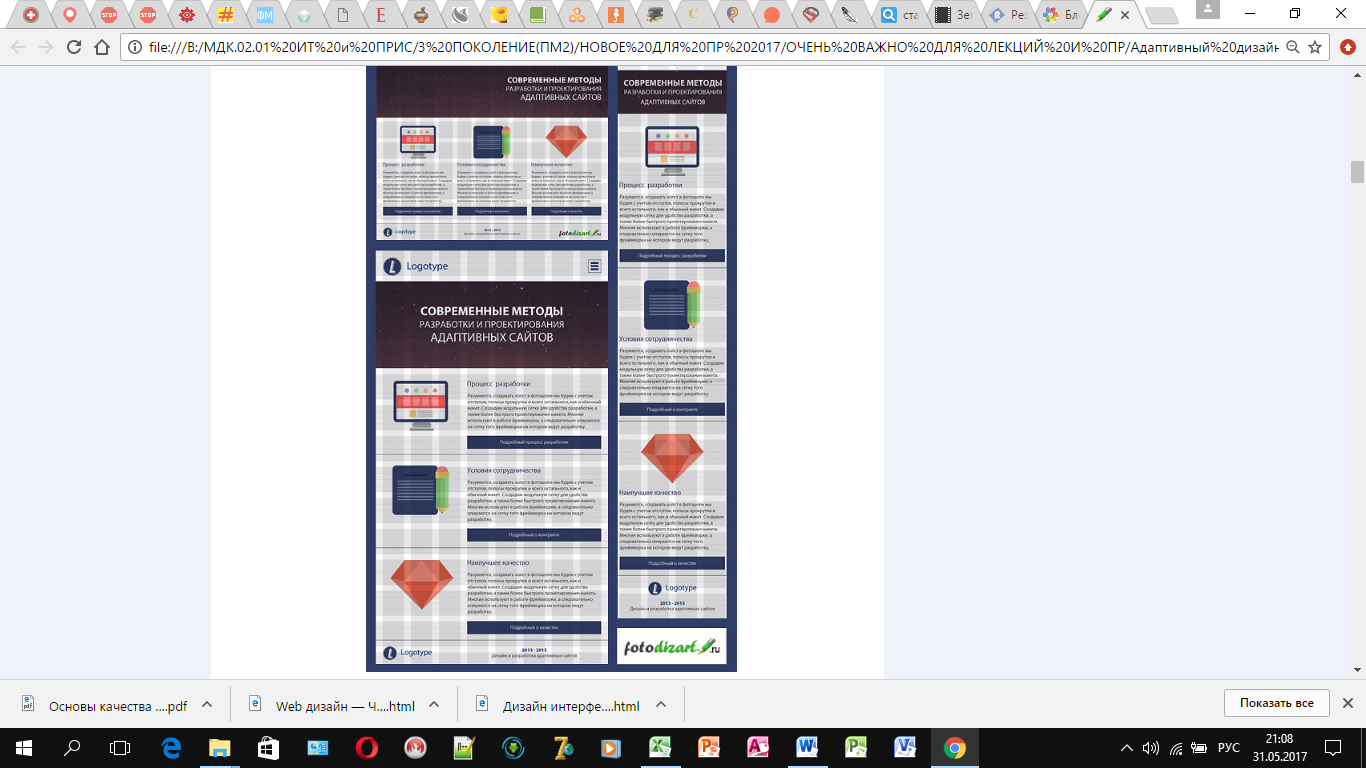


Рисунок 1

**Мediaquery и viewport в адаптивном дизайне**

Для того чтобы сообщить браузеру как отобразить размеры страницы и изменить ее масштаб используется метатег **viewport.**Данныйметатег прописывается в  сайта. Он позволяет разработчикам установить ширину экрана для устройств, которая прописана в css.

Записывается мета тег viewport так:

Рекомендуется записывать следующим образом:

* **width=device-width** – означает что ширина страницы сайта устанавливается в соответствие ширине экрана устройства.
* **initial-scale=1.0** – данный атрибут скажет браузеру установить соответствие масштаба 1:1 для пикселей, что означает не масштабировать.

Также для мета-тега могут устанавливаться другие атрибуты и параметры.

**MediaQueries**

В разработке адаптивных сайтов главную роль играют css3 **media queries** (медиа-запросы). Медиа-запросы включают в себя медиа-тип (принтеры, смартфоны, экраны, телевизоры, проекторы и др.) и условия, которое может принимать в свою очередь истину или ложь (true, false). В зависимости от того верный ли медиа-тип и выполняется ли  условие будут применяться различные стили css. Если будет верно, то будут применяться те стили, которые прописаны в этом медиа-запросе, если же будет ложным, то будут применяться обычные стили css.

Благодаря таким запросам и создаются различные отображения сайта, для мобильных, планшетов и экранов мониторов. Поддерживается всеми современными браузерами.

**Записывается следующим образом:**

***@media screen and (max-width: 1000px) {***

***.class {***

***свойство: значение;***

***}***

***}***

* **@media** – медиа-запрос;
* **screen** – медиа-тип (также называют тип носителя);
* **max-width: 1000px** – условие, которое должно выполняться (в нашем случае стили будут применяться, если ширина окна меньше ширины 1000px);
* **.class** – прописываются соответствующие селекторы (классы, id) в которых свойствам задаются новые значения.

В большинстве случаев для разработки адаптивного дизайна используются следующие медиа-функции.

* **max-width: width** – означает, что если ширина окна браузера меньше заданной ширины, то выполняется условие и применяются соответствующие стили (пример: ***max-width: 768px***, означает, что если ширина окна браузера меньше 768px то следует использовать стили, указанные в медиа-запросе).
* **min-width: width** – означает, что если ширина окна браузера больше заданной ширины, то условие выполняется, и применяются указанные стили в запросе (пример: ***min-width: 480px***).

**Нотакжемогутиспользоватьсяидругиефункции:** ***color, device-width, grid, height,* orientation: *landscape,* orientation: *portrait, resolution идругие.***

Значения, которые используются в медиа-функциях называют также  **breakpoints** (переломные или контрольные точки). В этих контрольных точках как Вы догадались, и меняется дизайн сайта.

* **320px** – мобильные
* **480px** – мобильные
* **768px** – планшеты
* **1024px** – планшеты, нетбуки
* **1280px** и более-запросы (пример: ***@mediaonlyscreenand (max-width: 1300px) { … }***).

***Mediaqueries прописываются в конце файла стилей, после всех основных стилей css.***

**Как сделать адаптивный дизайн сайта из фиксированного макета**

Пусть дана рабочая тема сайта с фиксированным размером, которую необходимо оставить и сделать из нее адаптивный макет, для этого нужно:

1 Сделать резервную копию шаблона (темы).

2 Далее потребуется программа для редактирования css, это может быть любой редактор кода, например brackets от adobe он бесплатный или Notepad++.

3 А также браузер googlechrome, с его инспектором кода (вызывается клавишей F12)

Далее начнем редактирование, для удобства можно перенести шаблон на денвер (локальный сервер), либо на поддомен, кому как удобнее, чтобы пользователи не видели все те правки, которые будут происходить с макетом. Первое что нужно сделать добавить мета-тег ***viewport.*** Далее необходимо **перевести все статические единицы измерения в относительные единицы измерения.**

Таковыми являются **px, необходимо перевести их в %, а шрифты задать в em.** В основном это касается **ширин и шрифтов.**

Ширину основного контейнера ***wraper {max-width: 960px;}*** оставляем без изменения, если записано ***width*** измените на ***max-width***. Для остальных контейнеров будем менять ширину на % отношение. Переводить мы будем по формуле:

**Размер контейнера (px) / размер основного контейнера (родителя) в (px) \* 100% = результат (%)**

**Например, статичный контейнер тела сайта 720px размер основного контейнера (его родителя), к примеру, стандартный 960px, тогда получим следующее 720/960\*100=75% Таким образом, мы переведем нашу статичную верстку в резиновую.**

Далее переведем наши шрифты, если они в **px** в **em**для этого опять же воспользуемся формулой:

**Размер шрифта (px) /16px (стандартный размер) = размер шрифта (em)**

**Например, размер шрифта 32рх, тогда 32/16=2em.**

После чего сделаем **изображения адаптивными.** Для этого в css пропишем следующие свойства и значения.

***img{***

***max-width:100%;***

***height: auto;***

***}***

Этот способ неплохо адаптирует картинки, единственное он не уменьшает вес картинок, а это значит, что на мобильных устройствах будут загружаться картинки с большим весом. Чтобы такого не было необходимо грузить разные картинки в зависимости от размера экрана.

После внесения всех этих изменений, если Вы внесли все верно, то на сайте ничего не должно измениться, а если попробовать уменьшить окна браузера дизайн и картинки должны сжиматься.

Итак, основа готова, теперь необходимо определить контрольные точки, в которых макет будет перестраиваться, продумать, как будут вести себя блоки, что будем скрывать и записать это все в медиа-запросы. Определять эти самые переломные точки мы будем при помощи браузера googlechrome.

Откроем в нем макет, нажмем клавишу **F12**и будем уменьшать размер окна браузера. В верхнем правом углу будут писаться размеры окна, нас интересует первое значение, оно указывает на ширину. Необходимо сжимать до тех пор, пока дизайн не перестанет корректно отображаться. В том размере, когда макет выглядит не корректно, и будем создавать контрольную точку. Определив первую переломную точку, запишем ее в файл **style.css** в самый конец после всех основных стилей. Допустим макет, имеет левый сайдбар и контентную часть с анонсами, и при 910px сайт начинает отображаться не корректно. В данном случае запишем следующий медиа запрос.

***@media only screen and (max-width: 910px){***

***/\* контентную часть мы делаем во всю ширину, отменяем выравнивание\*/***

***section {***

***width: 100%;***

***float: none;***

***}***

***/\* сайдбар мы делаем также во всю ширину, отменяем выравнивание\*/***

***aside{***

***width: 100%;***

***float: none;***

***}***

***}***

Если у этих блоков есть какие-то отступы (;  тогда ширину запишет следующим образом **width: 96%;**.

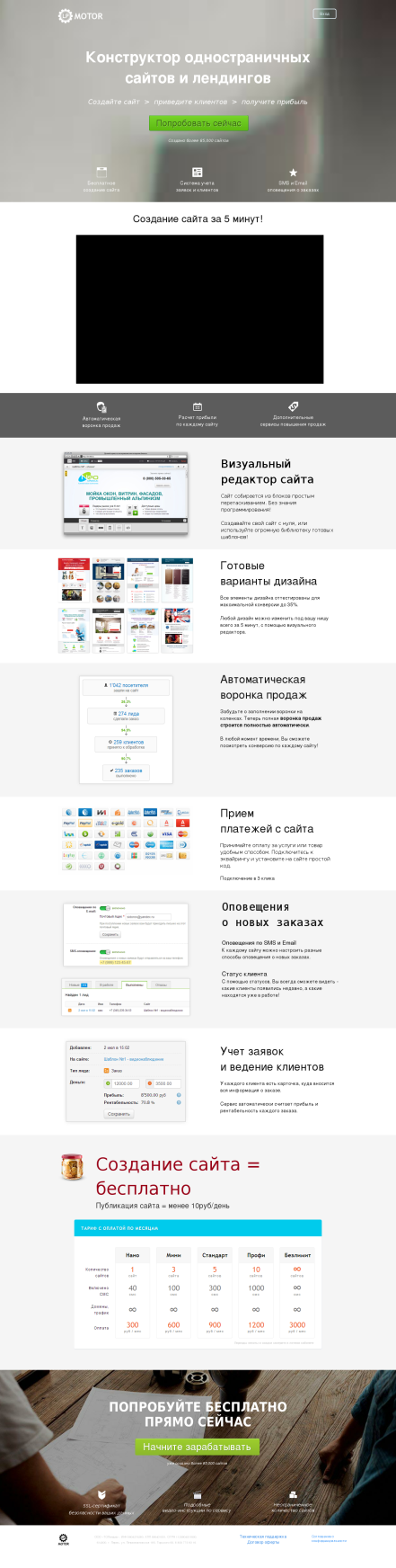
Таким образом, макет адаптирован на разрешение меньше 910px. Если окно браузера будет меньше 910px, то контент станет во всю ширину, а сайдбаротпустится под контентную часть и станет также во всю ширину.

По такому же принципу создаются и остальные контрольные точки, находим ширину, при которой макет ломается (смотрится не корректно) записываем медиа-запрос, формируем для него стили, задаем ширину блоков, можем скрыть менее важные блоки (**display: none**).

Как Вы заметили, в данном примере контрольные точки не будут жестко привязываться к размерам экранов, что в свою очередь гарантирует хорошее отображение сайта в различных устройствах, не зависимо от их разрешения.

**Ход работы**

**Задание.** Создать адаптивный макет посадочной страницы под названием «360».



**Контрольные вопросы**

1. Что такое верстка?
2. Какие бывают виды верстки?
3. Какие Вы знаете методы верстки?
4. В чем особенность адаптивной верстки?
5. Как реализуется адаптивная верстка?