**Адаптивный дизайн**

**План:**

1. [Введение в адаптивный дизайн](https://metanit.com/web/html5/11.1.php)
2. [Метатег Viewport](https://metanit.com/web/html5/11.2.php)
3. [Media Query в CSS](https://metanit.com/web/html5/11.3.php)
4. **Введение в адаптивный дизайн**

Сейчас все большее распространение находят различные гаджеты - смартфоны, планшеты, "умные часы" и другие устройства, которые позволяют выходить в интернет, просматривать содержимое сайтов.

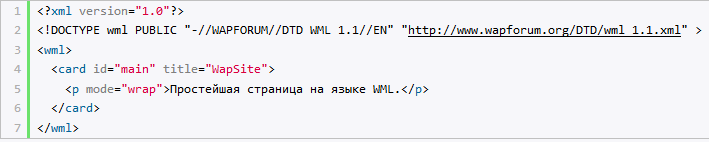
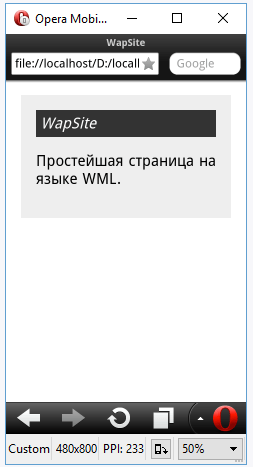
По некоторым оценкам уже чуть ли не половина интернет-траффика генерируется подобными гаджетами, разрешение экрана которых отличается от разрешения экранов стандартных компьютеров.

Подобное распространение гаджетов несет новые возможности по развитию веб-сайтов, привлечения новых посетителей, продвижению информационных услуг и т.д. Но вместе с тем появляются и новые проблемы.

Главная проблема заключается в том, что стандартная веб-страница будет по разному выглядеть для разных устройств с разным разрешением экрана. Первоначальным решением данной проблемы было создание специальных версий для мобильных устройств.

На заре распространения мобильных телефонов пользователи могли через телефон по протоколу WAP получать доступ к специальным wap-сайтам, которые были написаны на языке wml - языке на основе xml, похожем на html.

К примеру, простейшая веб-страница на этом языке разметке могла иметь следющий код:



Визуально она выглядела так:

Однако развитие самих гаджетов, их возможностей привело к тому, что сейчас мобильные телефоны представляют куда большие возможности по получению и отображению содержимого сайтов, а в написании подобных сайтов используется те же HTML5 и CSS3, что и для обычных сайтов.

Кроме того, появление все большего количества разнообразных устройств привело к тому, что веб-страницы необходимо подстраивать не только под небольшие экраны смартфонов или планшетов, но и под огромные экраны полноформатных широкоэкранных телевизоров или гигантских планшетов типа Surface Hub, которые также могут иметь доступ к интернету.

И для решения проблемы совместимости веб-страниц с самыми различными разрешениями самых различных устройств возникла концепция **адаптивного дизайна.**

Ее суть заключается в том, чтобы должным образом масштабировать элементы веб-страницы в зависимости от ширины экрана.

Хотя нередко до сих пор можно встретить ситуацию, когда для сайта создается отдельная мобильная версия, часто с префиксом ***m***, например, ***m.vk.com***.

Однако концепция адаптивного становится все более распространенной и доминирующей.

### **Тестирование адаптивного дизайна**

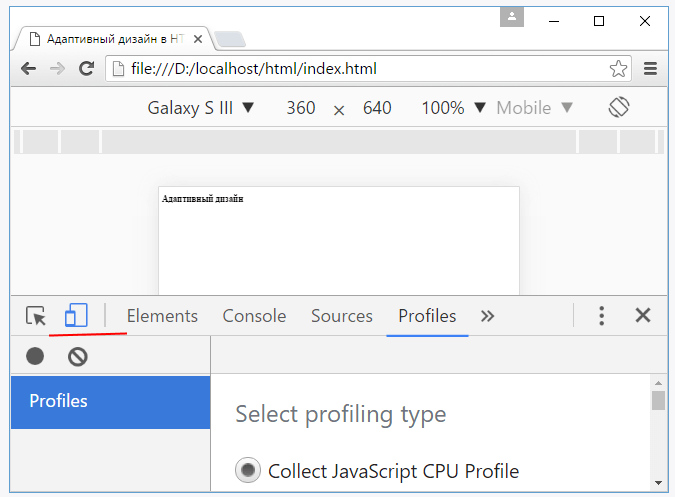
При разработке адаптивных веб-страниц можно столкнуться с трудностями тестирования, так как, как правило, разработка идет на обычных компьютерах.

Но многие современные браузеры позволяют эмулировать запуск веб-страницы на том или ином устройстве с различной шириной экрана.

Например, в Google Chrome надо перейти в меню

***Дополнительные инструменты -> Инструменты разработчика***.

После открытия панели разработчика в начале меню самой панели можно нажать на иконку мобильного телефона, и после этого можно будет эмулировать отображение страницы на различных устройствах – от небольших телефонов до широкоформатных телевизоров:

 В данном случае отображается веб-страница как она бы выглядела на устройстве Samsung Galaxy S III.

Но при желании можно выбрать другое устройство, либо даже создать эмуляцию какого-то нового устройства, которого нет во встроенном списке.

Подобные инструменты есть и в других современных веб-браузерах. Например, в Mozilla Firefox для их открытия надо перейти в меню ***Разработка -> Адаптивный дизайн***

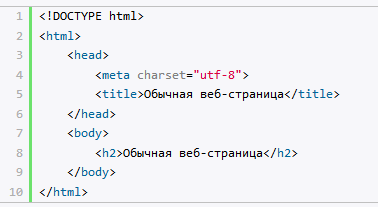
Еще одно решение заключается в использовании эмуляторов мобильных устройств.

Небольшой список подобных эмуляторов можно найти по следующему адресу: <http://www.mobilexweb.com/emulators>.

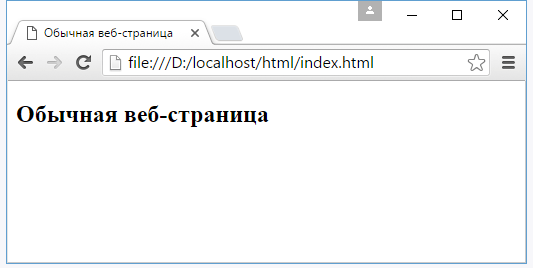
Наиболее популярным является эмулятор браузера Opera Mobile, который можно найти по адресу <http://www.opera.com/ru/developer/mobile-emulator>.

## 2. Метатег Viewport

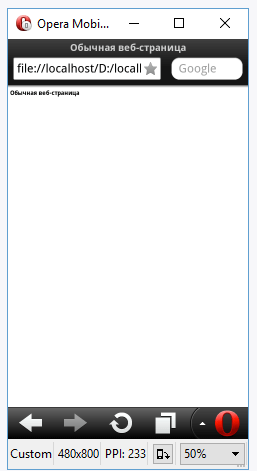
Рассмотрим один из ключевых моментов применения адаптивного дизайна – метатег ***viewport*** (по сути с этого тега и начивается адаптиный дизайн). Пусть для начала у нас есть следующая веб-страница:



Это стандартная веб-страница, которая в обычном браузере будет выглядеть следующим образом:



Однако если запустить эту же самую веб-страницу в эмуляторе мобильного устройства или на реальном мобильном устройстве, то с трудом сможем прочитать, что же на ней написано:



Применяя масштабирование, пользователь может наконец-то увидеть, что же там все таки написано. Однако это не очень удобно. При этом веб-страница имеет много пустого места, что не очень красиво.

Почему так происходит? Дело в том, что каждый мобильный браузер задает странице некоторые начальные размеры и потом пытается приспособить под размеры экрана текущего мобильного устройства.

Вся видимая область на экране браузера описывается понятием ***Viewport***.

По сути ***viewport*** представляет область, в которую веб-браузер пытается "впихнуть" веб-страницу.

Например, браузер Safari на iPone и iPod определяет ширину ***viewport*** по умолчанию равной *980* пикселям. То есть, получив страницу и вписав в ***viewport*** с шириной 980 пикселей, браузер сжимает ее до размеров мобильного устройства.

Например, если ширина экрана смартфона составляет *320* пикселей, то до этого размера потом будет сжата страница. И ко всем элементам страницы будет применен коэффициент масштабирования, равный *320/980*.

Почему в данном случае используется именно *980* пикселей, а, скажем, не реальный размер экрана?

Все дело в том, что по умолчанию браузер считает, что каждая веб-страница предназначена для десктопов. А обычной шириной десктопного сайта можно считать *980* пикселей.

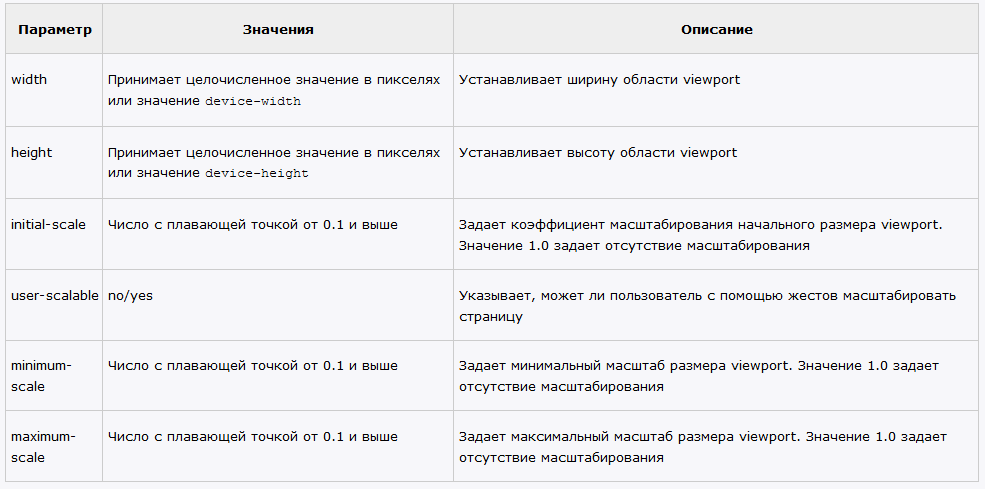
При этом для каждого браузера устанавливается своя ширина области ***viewport***, необязательно *980* пикселей.

Другие браузеры могут поддерживать в качестве начальной ширины другие значения. Но они также будут выполнять масштабирование.

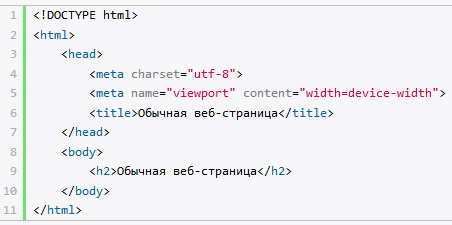
Чтобы избежать подобной не очень приятной картины, следует использовать метатег ***viewport***. Он имеет следующее определение:



В атрибуте ***content*** мета-тега можно определить следующие параметры:



Теперь изменим предыдущий пример – веб-страницу, использовав метатег:



|  |  |
| --- | --- |
|  | Веб-страничка выглядит лучше благодаря использованию метатега **viewport**.  Используя параметр ***width=device-width*** мы говорим веб-браузеру, что в качестве начальной ширины области ***viewport*** необходимо считать не *980* пикселей или какое-то другое число, а непосредственную ширину экрана устройства.  Поэтому затем веб-браузер не будет проводить никакого масштабирования, так как у нас ширина ***viewport*** и ширина одинаковы. |

Также можно использовать другие параметры, например, запретить пользователю масштабировать размеры страницы:



## Media Query в CSS

Другим важным элементом в построении адаптивого дизайна являются правила ***Media Query***, которые позволяют определить стиль в зависимости от размеров браузера пользователя.

В CSS2 уже было решение в виде правила ***media***, которое позволяет указать устройство, для которого используется данный стиль:



Правило ***media="handheld"*** указывает, что стили в ***mobile.css*** будут применяться к мобильным устройствам, в то время как правило ***media="screen"*** применяется к десктопным браузерам.

Однако многие современные мобильные браузеры по умолчанию считают, что страница предназначена для десктопов, поэтому в любом случае применяет правило ***media="screen"***. Поэтому на подобное решение не стоит полагаться.

Для решения этой проблемы в ***CSS3*** были введен механизм ***CSS3 Media Query***.

Например, чтобы применить стиль только к мобильным устройствам, можно написать так:



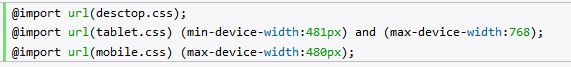
Значение атрибута ***media*** (*max-device-width:480px*) говорит о том, что стили из файла *mobile.cs*s будут применяться к тем устройствам, максимальная ширина экрана которых составляет 480 пикселей – то есть фактически это и есть мобильные устройства.

С помощью ключевого слова ***and*** можно комбинировать условия, например:

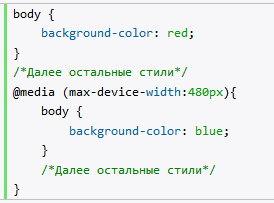


Данный стиль будет применяться, если ширина браузера находится в диапазоне от *481* до *768* пикселей.

С помощью директивы ***@import*** можно определить один *css-*файл и импортировать в него стили для определенных устройств:



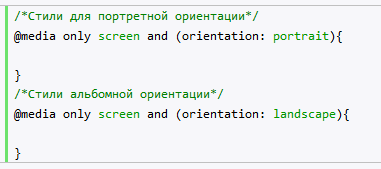
Также можно не разделять стили по файлам, а использовать правила ***CSS3 Media Query*** в одном файле css:



*Применяемые функции в CSS3 Media Query:*

* ***aspect-ratio***: отношение ширины к высоте области отображения (браузера)
* ***device-aspect-ratio***: отношение ширины к высоте экрана устройства
* ***max-width/min-width*** и ***max-height/min-height***: максимальная и минимальная ширина и высота области отображения (браузера)
* ***max-device-width/min-device-width***  ***max-device-height/min-device-height***: максимальная и минимальная ширина и высота экрана мобильного устройства
* ***orientation***: ориентация (портретная или альбомная)

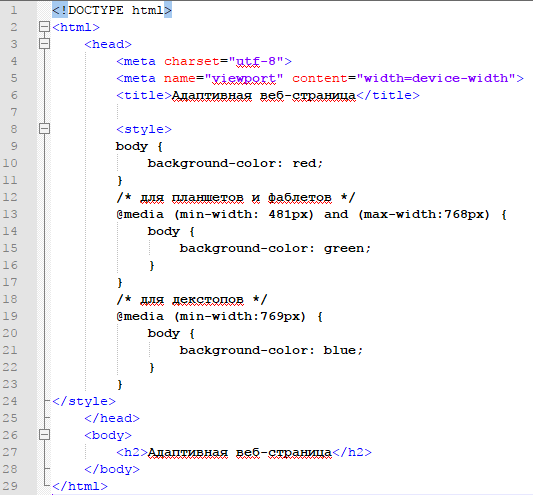
Например, можно задать разные стили для разных ориентаций мобильных устройств:

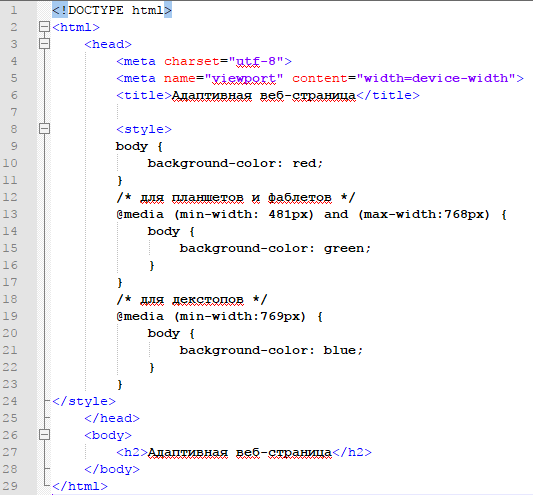


Таким образом, мы меняем лишь определение стилей в зависимости от устройства, а сами стили ***css*** по сути остаются теми же, что используем для создания обычных сайтов.

Как правило, при определении стилей предпочтение отдается стилям для самых малых экранов - так называемый подход ***Mobile First***, хотя это необязательно.

Например, определим следующую веб-страницу:





Сначала идут общие стили, которые актуальны прежде всего для мобильных устройств с небольшими экранами.

Затем идут стили для устройств с экранами средней ширины: фаблеты, планшеты. И далее идут стили для десктопов.

И например, на эмуляторе Opera Mobile при эмуляции устройства с шириной в 480 пикселей страница приобретет красный цвет:



А в браузере обычного компьютера страница будет иметь синий цвет, как и определено в стилях:

