**Адаптивная верстка сайта** — создание страниц сайта, автоматически подстраивающихся под размер, разрешение и ориентацию экрана устройства.

Адаптивная верстка позволяет просматривать контент без скролла страницы, а характерная для десктопа полоса прокрутки в правой части страницы просто отсутствует.

Адаптивность — важнейшее требование, которое предъявляется к современному сайту, особенно в текущих условиях поисковой оптимизации.

**8 принципов адаптивной верстки**

При адаптации сайта веб-разработчики стремятся к сохранению исходной концепции, сути проекта и прямого назначения ресурса. Также им необходимо обеспечить функциональность сайта и добавить некоторые особенности, которые позволят в дальнейшем производить подстройку под инструмент воспроизведения.

* **Прочность версии**

Отображение сайта на экране гаджета должно иметь сходство с отображением на экране ПК. Однако расположение важных элементов должно быть компактным и выстроенным по вертикали, то есть идти потоком сверху вниз.

* **Соотношение единиц измерения**

Необходимо помнить, что одни и те же пиксельные разрешения отображаются неодинаково на разных мониторах – компьютера и, например, айфона. Чтобы избежать недочетов, связанных с этой особенностью, необходимо задавать масштаб и координаты, отталкиваясь от общей для всех экранов точки. На данный момент принято равняться на верхнюю границу монитора.

* **Применение контрольных точек**

Чтобы отдельные элементы страницы бесконтрольно не расползались по ней, их привязывают к конкретным объектам, чье положение будет статичным, пока не будет осуществлен вход с другого устройства. То есть эти объекты будут выступать в роли своеобразных узлов, на которые будут ориентированы другие детали.

* **Интервалы значений**

Для пикселей указывают максимальные и минимальные значения по ширине и высоте. Благодаря этому содержание страницы остается в определенных рамках, которые делают восприятие удобным. Контент занимает часть экрана или весь монитор, но отображается полностью, не растягиваясь.

* **Принцип вложенности объектов**

Если один объект в качестве своих свойств содержит другой объект, то лучше сохранить их взаимосвязь и поместить в общей контейнер, чтобы не тратить время на дополнительные настройки. В роли таких объектов выступают логотипы, кнопки и многие другие детали веб-страницы.

* **Корректность шрифтов**

Шрифт допустимо переводить в web-формат, чтобы он, ориентируясь на размеры монитора, автоматически менял свой размер. Но в этом случае для шрифта понадобится время на загрузку. Не все пользователи обладают терпением, чтобы дожидаться момента, когда подгрузится весь контент – адаптивный сайт должен открываться быстро.

* **Правильное использование графики**

Если при наполнении сайта используется графика, то продумайте, по какому принципу ее отображать. Если она содержит мелкие детали, то делайте ее растровой, если нет – то векторной. Также все картинки следует сжать таким образом, чтобы их формат могли поддерживать старые браузеры.

* **Корректные медиазапросы**

Медиазапросы задают:

* типы поддерживаемых устройств;
* условия для демонстрации контента.

Для каждого медиазапроса будут использоваться необходимые стили CSS-языка.

**Разрешения экранов для адаптивной верстки**

При разработке адаптивного сайта веб-дизайнеры учитывают не конечные размеры страницы, а конкретные пределы, превышение которых приводит к изменению формата отображения. Существуют общепринятые размеры для адаптивной верстки:

* для смартфонов — 320 px, 480 px и выше;
* для планшетов — 768 px и выше;
* для нетбуков — 1024 px и выше;
* для мониторов — 1280 px, 1600 px, 1920 px и выше.

При неадаптивной верстке элементы статично привязаны к ширине экрана, именно поэтому они увеличиваются/уменьшаются в размерах или изменяют свое положение. Такое поведение объектов будет напрямую зависеть от типа устройства, с помощью которого открывается сайт.

На сегодняшний день выполнение адаптивной верстки осуществляется с опорой на Mobile First. То есть разработка ведется с ориентацией на мониторы мобильных устройств и только потом дорабатывается настройками под десктопные мониторы.

**4 техники адаптивной верстки**

* **Flexible layout, или гибкий макет**

При этой технике происходит переход фиксированных значений в относительные. Для этого используется формула, определяющая процентное соотношение: target ÷ context = result. Другой способ – поделить ширину объекта на ширину экрана. Полученное число равно процентному значению ширины объекта, которое используется для его подстройки под размер экрана гаджета.

Например, полная ширина макета равна 1000 пикселям, но делится на два части: левая шириной 250 пикселей и правая, шириной 530 пикселей. Далее, используя принцип прямой пропорции, получаем 25 % для левой части, 53 % для правой.Использование такого способа создания макета позволяет получить сайт, который автоматически подстраивается под любые размеры экранов.

* **Flexible media, или гибкое изображение**

Любой вид медиаконтента адаптируют, применяя свойство {max-width: 100%}. Такой подход способствует изменению размеров картинок, видео и прочих видов медиа в соответствии с изменениями ширины экрана.

* **Flexible Embedded Media, или гибкое встроенное медиа**

Принцип соотношения max-width не даст результатов, если на странице сайта присутствует, например, ролик YouTube, то есть медиаконтент постороннего ресурса. Для корректного отображения подобного встроенного контента его нужно правильно разместить внутри родительского объекта. При этом родительский объект должен иметь параметры: ширина = 100 %, высота = 0. Тогда он будет менять свой масштаб в соответствии с размерами монитора.

Далее нужно задать отступ в нижней области родительского объекта. Чтобы соблюсти пропорциональность высоты и ширины, нужно установить значение отступа согласно соотношению сторон видео.Например, стороны видео соотносятся, как 16:9. Используем формулу процентного соотношения: 9/15=0,5625 или 56,25%.

* **Media queries, или медиазапросы**

Этот способ адаптирует сайт таким образом, что он при помощи CSS модуля приобретает способность подстраиваться под различные типы устройств. CSS устанавливает взаимосвязь между разрешением экрана и установленными стилями. Media queries находит широкое применение в мобильной и TV разработках, печати и в таких типах устройств, как 3D-очки.

Для упрощения действий лучше применять правило @media внутри имеющейся таблицы стилей.

Медиа-запросы — это правила-триггеры, применяемые к CSS, которые реагируют на тип устройства, тип экрана, отображаемый браузер или даже ориентацию устройства.

Чтобы сделать адаптивную верстку, нужно применять для этого процесса технологии: HTML5 и CSS3.

При использовании CSS3 появляется возможность задавать такие величины объектов, как нужный размер для конкретного разрешения монитора, процентная доля заполняемого пространства (например, 80 % рабочего пространства экрана) и прочие параметры.

Возможность создания элементов, подстраиваемых под заданное разрешение монитора предоставляет правило медиазапросов. Например:

«ь@media screen and (max-width: 600px) {

   /\* CSS-стили\*/;

}

*Логический оператор and - позволяет связывать вместе разные типы условий*

Медиа-запрос следует правилу, основанному на параметрах минимальной и/или максимальной ширины, наряду с другими факторами (например, разрешение, браузер и ориентация экрана).

**Относительность и размеры**

Адаптивная верстка предусматривает относительность буквально во всем. Например, в определении размера шрифта, размера отступов и конечно, ширины и высоты.

Относительные значения определяют 4 главных параметра:

1. Ширину (width);
2. Высоту (heights);
3. Межблоковое расстояние или отступы (margin);
4. Отступ от контента до границы блока (padding).

Все вышеуказанные параметры чаще всего прописываются в процентном соотношении. Также допустимо задействовать vw\* и vh\* (только внутри div-тега):

* **vw**(сокращенно от Viewport Width) – процентная величина от суммарной ширины области viewport’а (пользовательская область просмотра);
* **vh**(сокращенно от Viewport Height) — процентная величина от суммарной высоты viewport’а.

### Размерность шрифтов

Адаптивная верстка подразумевает использование 2 единиц — em (допускает вариативность для любого элемента документа) и rem (задается один раз на страницу). Em — это размер шрифта, который используется на смартфоне или другом мобильном устройстве по умолчанию. Прописываем 3em — получаем шрифт втрое больше, чем главный шрифт страницы для конкретного устройства.

**Размеры экранов для адаптивной верстки**

Контрольные точки по умолчанию принято использовать с каноническими значениями плотности пикселей:

* Мониторы full-hd — от 1920 px.
* Мониторы низкого класса (не full-hd) — от 1280 px.
* Мобильные устройства высокого класса (смартфоны) — от 320 px.
* Нетбуки — от 1024 px.
* Планшеты — от 768 px.

**Синтаксис**

Синтаксис подразумевает ввод медиазапроса через правило @media. Условия разделяется оператором (not, and, or), далее указываются параметры. Ширина, высота, ориентация страницы, соотношения разных элементов, либо разрешение экрана — все это параметры @media.

Для resolution (разрешение экрана) допустимо использовать любую из 3 единиц: dppx, dpi или dpcm (кол-во точек на пиксель / дюйм / сантиметр соответственно).

Для width разрешено задать минимальную и максимальную ширину. Аналогично — для heights (высоты) и aspect-ratio (форматного соотношения). Свойство CSS **aspect-ratio**позволяет создавать блоки с пропорциональными размерами, при этом высота и ширина блока автоматически рассчитываются как соотношение.

  aspect-ratio: 1 / 1;

  aspect-ratio: 16 / 9;

  aspect-ratio: 0.5;

*Образцы значений для aspect-ratio*

Соотношение ширины / высоты всегда указывается в дробном представлении, но только с целыми числами.

Стандартное значение orientation в адаптивной верстке — горизонтальное (альбомный лист) или вертикальное.  
Например:

  @media all (min-width: 1200px) {

  div {

  width: 800px

  }

  }

  @media all (min-width: 400px) {

  div {

  width: 350px

  }

  }

А этот код указывает ширину и высоту страницы:

  @media screen and (min-aspect-ratio: 239/100) {

  main {

     height: 80%;

  }

  }

Плотность пикселей, ориентация страницы задаются аналогично.

**Размер элементов в адаптивной верстке**

Аналогичным образом (при помощи указания минимального и максимального значения) прописываются размеры компонентов на странице. Взгляните на этот фрагмент:

  div {

  height: 70%;

  max-height: 1900px;

  }

Мы установили максимальную высоту для элемента 1900 px. Также обратите внимание на указание height в процентах.

В указании размера элементов нет ничего нового:

* max-height — максимальная высота.
* max-width — максимальная ширина.

С минимальными значениями все аналогично: min-height и min-width по высоте и ширине, соответственно.

Задав максимальные / минимальные размеры компонента, вы можете не беспокоиться о том, что экран пользователя слишком маленький для комфортного просмотра элемента. Такой элемент в любом случае не будет меньше минимального значения

**Адаптивная верстка изображений**

До сих пор мы говорили об элементах документа обобщенно. Пусть это и очевидно, но необходимо упомянуть правила верстки изображений. Чтобы создать адаптивное изображение, задайте максимальную ширину элемента — например, 100 %:

  media {max-width 100%;) {

Таким образом мы уточнили: максимальная ширина картинки должна составляет 100% от пользовательского экрана — например, браузерного окна.

Адаптивные изображения могут быть весьма эффективными, но есть у них и недостаток: физический размер файла не уменьшается. Поэтому не стоит ожидать значительного уменьшения скорости загрузки страницы даже при глобальной замене обычных изображений на адаптивные.

}} **Методы разработки адаптивной верстки**

На первом этапе процесса адаптивной верстки нужно создать изображение, которое будет растягиваться в соответствии с размерами монитора.

HTML

|  |
| --- |
| <div>  <img src= «../images/1.jpg»>  </div> |

Здесь все просто: это контейнер, чтобы «упаковать» изображение.

CSS

|  |
| --- |
| div {  width: 100%;  text-align: center;  }  div img {  width: 100%;  height: auto;} |

Здесь соотнесли ширину контейнера div и ширину изображения img, чтобы img меняло свои размеры в соответствии с изменениями размеров div.

То есть получена картинка, которая будет подстраиваться под любое разрешение экрана (неважно, 320\*240 px или 1920\*1080 px), занимая весь монитор.

В приведенном примере итоговое изображение не будет иметь хорошего качества: на большом экране оно будет слишком растянуто, а на маленьком — слишком сжато.

Для доработки изображения нужно конкретизировать его ширину для минимальных и максимальных разрешений.

|  |
| --- |
| div {  margin: 0 auto;  width: 1000px;  max-width: 90%; /\* контейнер не превышает 90% ширины экрана \*/  min-width: 500px;  }  div img {  max-width: 100%;  height: auto;  } |

CSS

На этом этапе параметры «упаковочного» контейнера были откорректированы. Теперь его размеры будут меняться в интервале оптимальных значений 500-1000 px. Соответственно, и размеры картинки тоже будут изменять свою величину.

Так как при адаптивной верстке предполагается оптимизация всего содержимого сайта, то перейдем к более сложной процедуре — созданию мини-галереи.

Начнем с уже знакомых шагов.

|  |
| --- |
| <div class= «image\_gallery»>  <div>  <img src= «../images/1.jpg» title=»Деревья</em><em>, листья</em><em>…» />  </div>  <div>  <img src= «../images/2.jpg» title=»Зима</em><em>» />  </div>  <div>  <img src= «../images/3.jpg» title=»Озеро</em><em>» />  </div>  <div>  <img src= «../images/4.jpg» title=»Райский</em><em> остров</em><em>» />  </div>  </div> |

HTML

Здесь было создано несколько изображений, которые поместили в контейнер.

CSS

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | div.image\_gallery {  margin: 0 auto;  width: 1000px;  text-align: center;  max-width: 90%; /\* контейнер не превышает 90% ширины экрана \*/  min-width: 500px;  }  img {  float: left;  max-width: 48%;  height: auto;  padding: 1%; /\* небольшие оступы для изображений \*/  } |

По результату этих действий получена галерея, каждая из картинок которой будет менять свой размер, реагируя на изменяющийся размер контейнера. Для лучшего визуального восприятия изображения с каждой его стороны установлен небольшой отступ в 1 %.

(При помощи float можно размещать блочные элементы в строке.Элементы float:left двигаются влево, а если это невозможно, то вниз, автоматически адаптируясь под ширину контейнера, получается эффект, аналогичный display: inline-block, но с особенностями float. Актуально: float -> flexbox. )

Созданная в этом примере мини-галерея будет подстраиваться под мониторы с различными разрешениями. То сеть цель достигнута — адаптивная верстка выполнена верно.

**Инструменты для адаптивной верстки**

Для тех, кто занимается веб-дизайном и хочет улучшить свои умения, ниже приведены инструменты, способные облегчить процесс работы над адаптацией сайта.

* **Инструментарий Bootstrap**

Бесплатная платформа со свободным набором инструментов для создания сайтов и приложений. Bootstrap регулярно обновляется и используется для создания кнопок, меток, блоков навигации и прочих систем управления содержанием сайта.

Эта платформа — своеобразный скелет для адаптивных веб-сайтов. Нужно просто поэтапно сформировать структуру HTML и правильно классифицировать созданные объекты. Далее фронтенд отобразит готовые блоки с оформлением.

**Сервис Responsinator**

С помощью этого сервиса можно увидеть, как будет отображаться сайт на разных мониторах. Для этого достаточно ввести URL сайта в верхней панели сервиса. Отлично решает вопросы с доработкой адаптивности. Выбирая нужный девайс в предложенном списке, можно найти возможные ошибки в отображении страниц.

При проверке сайта на Responsinator демонстрируется версия, подстроенная под выбранное разрешение. Есть шанс, что в итоге на реальном устройстве изображение будет выглядеть несколько иначе, но по большей части отличия не существенны.

* **Инструмент Caniuse**

Еще один сервис, позволяющий свести к минимуму ошибки при адаптивной верстке. В Caniuse можно найти информацию о способности браузера поддерживать те или иные свойства сайта: например, CSS свойство sticky или формат шрифта ttf.

Caniuse идеально подойдет тем, кто доводит до совершенства адаптивную верстку и кроссбраузерность.

**Типичные ошибки при разработке адаптивной верстки**

Когда дело касается создания адаптивной версии интернет-магазина, то в большинстве случаев допускаются одни и те же ошибки. Они отражаются на результативности сайта к подстройке под экран, а также затрудняют действия пользователя. Основными такими ошибками будут:

* **Размеры шрифта и активных элементов**. Те объекты, которые хорошо видны на большом экране, становятся трудно воспринимаемыми на маленьких мониторах мобильных устройств. Чтобы не допустить такой ситуации, используйте шрифт не менее 12 пикселей. Также подбирайте корректные размеры для остальных важных элементов: кнопок, интерактивных деталей и т.д.
* **Блоки с прокруткой**. Чтобы сэкономить место, часто используют блоки, которые можно прокручивать. Если гаджет имеет сенсорный экран с небольшой шириной, то прокрутка приносит массу неудобств пользователю. В таких случаях лучше использовать функцию «раскрытия» блока кликом или просто показывать блок во весь его размер.
* **Объект с событием, реагирующим на касание**. На многих сайтах можно встретить элементы, запускающие событие, когда на него наводят курсор: например, всплывающие подсказки, меню и т. д. Нужно учитывать факт того, что на мобильных гаджетах курсор отсутствует. То есть то, что можно сделать на десктопе с помощью клика мыши, на сенсорных экранах приходится делать с помощью касания.
* **Несворачиваемое меню**. При адаптивной верстке создают удобные для пользователя детали. А массивное меню, раскрываясь на весь экран, явно не отвечает идеям адаптивности, так как просто-напросто мешает. Поэтому следует предусмотреть иконку, которая при необходимости скрывала бы или раскрывала меню.
* **Отсутствие глубокого тестирования**. Перед тем как запустить сайт интернет-магазина, нужно обязательно провести проверку его функциональности, а также уделить внимание основным путям посетителей в разных браузерах. Это поможет увидеть допущенные при разработке ошибки.
* **Долгая загрузка страниц сайта**. Каждый интернет-пользователю ценит скорость работы сайта. Так как она напрямую зависит от «веса» страницы, то нужно сделать объем составляющих страницу элементов минимальным. Чтобы добиться этого, содержание распределяют по дополнительным вкладкам, снимая нагрузку с основной части. Изображения, Java-скрипты и CSS-файлы тоже играют роль в скорости загрузки. Поэтому при разработке сайта нужно оставлять только важную информацию. А использование специальных инструментов поможет снизить размер ресурсных файлов.
* **Отсутствие предварительного анализа аудитории пользователей**. Это важный этап, который нельзя игнорировать. Цель анализа — понять, какими гаджетами для выхода в Интернет пользуется аудитория и какие действия предпринимают пользователи во время веб-серфинга. Только зная ответы на эти вопросы, можно разработать сайт, способный удовлетворить требования аудитории.

### На какие размеры экрана нужно ориентироваться

При составлении медиазапросов нужно ориентироваться на так называемые **переломные (контрольные) точки дизайна**, т.е. такие значения ширины области просмотра, в которых дизайн сайта существенно меняется, например, появляется горизонтальная полоса прокрутки. Чтобы определить эти точки, нужно открыть сайт в браузере и постепенно уменьшать область просмотра.

/\* Smartphones (вертикальная и горизонтальная ориентация) ----------- \*/

@media only screen and (min-width : 320px) and (max-width : 480px) {

/\* стили \*/

}

/\* Smartphones (горизонтальная) ----------- \*/

@media only screen and (min-width: 321px) {

/\* стили \*/

}

/\* Smartphones (вертикальная) ----------- \*/

@media only screen and (max-width: 320px) {

/\* стили \*/

}

/\* iPads (вертикальная и горизонтальная) ----------- \*/

@media only screen and (min-width: 768px) and (max-width: 1024px) {

/\* стили \*/

}

/\* iPads (горизонтальная) ----------- \*/

@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: landscape) {

/\* стили \*/

}

/\* iPads (вертикальная) ----------- \*/

@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: portrait) {

/\* стили \*/

}

/\* iPad 3\*\*\*\*\*\*\*\*\*\*/

@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: landscape) and (-webkit-min-device-pixel-ratio: 2) {

/\* стили \*/

}

@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: portrait) and (-webkit-min-device-pixel-ratio: 2) {

/\* стили \*/

}

/\* Настольные компьютеры и ноутбуки ----------- \*/

@media only screen and (min-width: 1224px) {

/\* стили \*/

}

/\* Большие экраны ----------- \*/

@media only screen and (min-width: 1824px) {

/\* стили \*/

}

/\* iPhone 4 ----------- \*/

@media only screen and (min-width: 320px) and (max-width: 480px) and (orientation: landscape) and (-webkit-min-device-pixel-ratio: 2) {

/\* стили \*/

}

@media only screen and (min-width: 320px) and (max-width: 480px) and (orientation: portrait) and (-webkit-min-device-pixel-ratio: 2) {

/\* стили \*/

}

/\* iPhone 5 ----------- \*/

@media only screen and (min-width: 320px) and (max-height: 568px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 2){

/\* стили \*/

}

@media only screen and (min-width: 320px) and (max-height: 568px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 2){

/\* стили \*/

}

/\* iPhone 6 ----------- \*/

@media only screen and (min-width: 375px) and (max-height: 667px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 2){

/\* стили \*/

}

@media only screen and (min-width: 375px) and (max-height: 667px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 2){

/\* стили \*/

}

/\* iPhone 6+ ----------- \*/

@media only screen and (min-width: 414px) and (max-height: 736px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 2){

/\* стили \*/

}

@media only screen and (min-width: 414px) and (max-height: 736px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 2){

/\* стили \*/

}

/\* Samsung Galaxy S3 ----------- \*/

@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 2){

/\* стили \*/

}

@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 2){

/\* стили \*/

}

/\* Samsung Galaxy S4 ----------- \*/

@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 3){

/\* стили \*/

}

@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 3){

/\* стили \*/

}

/\* Samsung Galaxy S5 ----------- \*/

@media only screen and (min-width: 360px) and (max-height: 640px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 3){

/\* стили \*/

}

@media only screen and (min-width: 360px) and (max-height: 640px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 3){

/\* стили \*/

}

CSS

Чтобы проверить, как выглядит ваш сайт на разных устройствах, можно воспользоваться сервисом [Responsive design testing for the masses](http://mattkersley.com/responsive/" \t "_blank).

### 7. Стратегии использования медиа-запросов

Для создания дизайна, позволяющего лучшим образом отображать сайт на различных устройствах, используют общие стратегии медиа-запросов:  
1) Уменьшение количества колонок (столбцов) и постепенная отмена обтекания для мобильных устройств.  
2) Использование свойства max-width вместо width при задании ширины блока-контейнера.  
3) Уменьшение полей и отступов на мобильных устройствах (например, нижних отступов между заголовком и текстом, левого отступа для списков и т.п.).  
4) Уменьшение размеров шрифтов для мобильных устройств.  
5) Превращение линейных навигационных меню в раскрывающиеся.  
6) Скрытие второстепенного содержимого на мобильных устройствах с помощью display: none.  
7) Подключение фоновых изображений уменьшенных размеров.