

АДАПТИВНЫЙ И ОТЗЫВЧИВЫЙ ДИЗАЙН ДЛЯ СОЗДАНИЯ ВЕБ-САЙТОВ

С.Л.Гончаров

В современном мире, чтобы создать сайт необходимо использовать адаптивный или отзывчивый шаблон. Для разработки подобных шаблонов используются фреймворки, наиболее популярным из которых является Bootstrap от компании Twitter. Шаблоны в основном используются в таких популярных системах управления контентом, как Joomla! и WordPress. В работе рассматриваются основные концепции адаптивного и отзывчивого дизайна, а так же правила и условия их использования.

Введение

Современные технологии развиваются невероятно быстрыми темпами. Сегодня мы имеем возможность зайти в интернет со стационарного компьютера, ноутбука и телефона, а также практически с любого домашнего прибора, подключённого к компьютерной сети. Разработчики веб-дизайна сайтов и веб-приложений с трудом адаптируются к этим изменениям.

В начале эры веб-дизайна, когда большинство пользователей выходили в интернет с компьютеров, имевших мониторы с одинаковым разрешением экрана, делать дизайн для сайтов было просто. Достаточно было сделать обычный статический дизайн под фиксированную ширину экрана и быть уверенным, что подавляющее большинство пользователей увидит сайт так, как и задумано.

Что же делать, чтобы пользователи, пришедшие к вам, могли увидеть сайт в подходящем виде и не испытывали проблем с навигацией по нему? Разработчики сайтов стали искать для этого разные решения. Первое решение, которое пришло в голову всем, когда появились мобильные устройства с доступом в интернет - создать просто отдельную «лёгкую» версию сайта для мобильных устройств. По такому пути пошли многие крупные порталы. Однако сейчас это решение уже не кажется таким привлекательным из-за резкого роста различных видов и подвидов устройств, с которых пользователи выходят в сеть. При этом владельцу планшета или смартфона тоже часто хочется посмотреть полноценную, а не урезанную версию сайта. Поэтому, в конце концов, дизайнеры и веб-разработчики пришли к идее, что нужно делать один вариант сайта, но такой, который бы одинаково хорошо отображался на любых устройствах. С этой идеи и берёт начало «адаптивный» дизайн. Но вначале о первых попытках адаптации дизайна сайта к новым условиям.

Резиновый дизайн

Резиновый дизайн появился довольно давно, долгое время соседствуя со статическим, фиксированным дизайном. Основная его особенность в том, что ширина колонок задаётся в процентах, а не в пикселях, поэтому веб-страница при любом разрешении монитора растягивается на всю ширину экрана. Сделать

такой дизайн, особенно при использовании блочной вёрстки несколько сложнее, чем фиксированный, зато пользователь не будет видеть пустых полей на странице или возникающих полос прокрутки по краям окна. Примечательно, что резиновый дизайн изначально довольно широко стал использоваться в российском интернете («рунете»), в то время как на Западе разработчики очень долго сохраняли приверженность статическому дизайну.

Тем не менее, у простого резинового дизайна есть определённые ограничения и недостатки. При таком большом разбросе размеров экрана, который есть у современных устройств, сложно реализовать резиновый дизайн таким образом, чтобы при любом разрешении всё одинаково хорошо выглядело, чтобы содержимое веб-страницы не «разъезжалось», чтобы картинки и видео не вылезали за границы блоков и т.п. По этой причине для создания действительно универсального сайта потребовались другие решения, в частности создание «лёгкой» версии сайта, о которой говорилось выше.

Адаптивный дизайн

Желая, чтобы сайт одинаково хорошо выглядел и на мобильном телефоне и на большом мониторе, разработчики некоторых сайтов пошли по пути разработки нескольких, по сути, независимых, вариантов дизайна для разных устройств. Когда пользователь заходил на сайт, сервер определял тип устройства и разрешение экрана и отдавал ту или иную версию сайта. При этом каждая версия была заранее оптимизирована под то или иное разрешение экрана. Для версий под экраны с маленьким разрешением использовались уменьшенные изображения, а некоторые элементы вообще убирались. Сама компоновка блоков, естественно, также отличалась от «большой» версии.

Отзывчивый дизайн

Последним методом, к которому пришли разработчики, стал отзывчивый дизайн (англ. responsive design), который противопоставили обычному «адаптивному» дизайну (англ. adaptive design), хотя до настоящего времени эти термины часто путают и употребляют в похожем смысле. Отзывчивый дизайн соединяет в себе черты адаптивного и резинового дизайнов - он автоматически подстраивается под любую ширину экрана, как и резиновый, при этом, в процессе подстраивания может меняться расположение блоков, размер картинок, могут появляться, пропадать или изменяться некоторые элементы (например, обычное горизонтальное меню может поменяться на выпадающее и т.п.).

Данный подход впервые был предложен еще в начале 2009 года Итаном Маркоттом, который впоследствии опубликовал целую книгу «Responsive Web Design» [1].

Конечно, отзывчивый дизайн выглядит впечатляюще и привлекателен по сравнению с адаптивным тем, что его можно сделать в единственном варианте. Однако и этот вариант дизайна имеет свои недостатки. Обычно в нём используются новые конструкции языка CSS, которые не поддерживаются рядом старых устройств, кроме того, «универсальный» дизайн часто получается излишне громоздким. Файлы, содержащие контент страницы, довольно

большого размера, при этом даже небольшое устройство (например, смартфон) вынуждено будет загрузить их целиком, а скрипты, подстраивающие дизайн под нужное разрешение, могут замедлить работу такого небольшого устройства.

Основные принципы отзывчивого дизайна

Отзывчивый дизайн — отличное решение проблемы корректного отображения сайта на разных экранах. Однако разработчикам, привыкшим к традиционным методам создания сайтов, зачастую трудно понять основы. С каждым днём появляется всё больше различных устройств, имеющих разные размеры экрана, поэтому создание дизайна в пикселях и только для настольных компьютеров/смартфонов остаётся в прошлом. Основные принципы отзывчивого дизайна — дизайна, совмещающего в себе адаптивность и резиновость заключаются в следующем.

Поток. Когда размер экрана уменьшается, содержимое страницы начинает занимать больше высоты, и элементы начинают смещаться вниз. Такое поведение называется *потоком*. Это было недоступным при использовании фиксированных размеров элементов веб-страницы.

Относительные единицы измерения. Область просмотра страницы может быть монитором, экраном мобильного и т.п.. Плотность пикселей на разных экранах также разная, поэтому нам нужны гибкие единицы измерения, работающие везде. Отзывчивый дизайн - именно тот случай, когда относительные единицы измерения вроде процентов становятся действительно полезными.

Контрольные точки (Breakpoints). Контрольные точки позволяют менять расположение блоков на странице только в случае использования экрана с определёнными размерами. Например, на настольных компьютерах у сайта будет три колонки, а на мобильных — только одна. Как вы расставите контрольные точки, зависит от поведения контента. Если содержимое страницы неверно отображается («ползёт»), то следует добавить контрольную точку.

Максимальные и минимальные значения. Контент, занимающий всю ширину экрана — это хорошо, если он отображается на мобильном. Если же страница открывается на большом экране, то появляются пустые места, картинки начинают отображаться не на тех местах, где задумано. Поэтому предлагается использовать минимальные и максимальные значения размера страницы. Например, если задать блоку свойства свойства ширины — 100%, а максимальную ширину установить в 1000 пикселей, то он будет отображаться на весь экран, если ширина экрана меньше 1000 пикселей, в противном случае, блок будет занимать 1000 пикселей.

Вложенные объекты. Если на странице будет много элементов, зависящих от расположения других элементов, то их будет тяжело контролировать. Лучше использовать один контейнер для таких элементов. Теперь уже оправдано использование статичных единиц измерения вроде пикселей. Они полезны для содержимого, которое не хотим адаптировать к размеру экрана - например, это может быть логотип или кнопка.

Где изначально открывается сайт (монитор или мобильный телефон).

С технической стороны нет никаких отличий - можно писать базовую разметку для мобильных и расставлять контрольные точки для десктопов (mobile first) или наоборот.

Веб-шрифты или системные шрифты. Можно использовать веб-шрифты. Они и выглядят красиво, но не стоит забывать, что каждый подключённый шрифт будет загружаться с удалённых серверов. Соответственно, чем больше шрифтов, тем медленнее загружается страница. С другой стороны, системные шрифты загружаются моментально за исключением случаев, когда пользователь не имеет локально установленного шрифта, используемого на странице. В таких случаях браузер будет использовать шрифт по умолчанию.

Растровые или векторные изображения. Если изображение, используемое на странице много множество мелких, то используйте растровый формат. В противном случае используйте векторный формат. Для растровых изображений используйте форматы jpg, png или gif, для векторных лучшим выбором будут и svg и ico. Каждый из форматов имеет свои преимущества и недостатки. В любом случае, всегда лучше помнить о размере изображений - ни одна картинка не должна попасть в онлайн, не будучи оптимизированной (сжатой). Векторные изображения зачастую избавлены от лишнего размера, однако они не поддерживаются старыми браузерами. Также стоит помнить, что если векторное изображение содержит много деталей, то оно может весить больше растрового.

Общие требования к структуре страниц отзывчивого дизайна

1. Ширина контейнера страницы должна быть резиновой (разрешается зафиксировать максимальный размер).

2. Нельзя использовать жесткую модульную сетку, например, некоторое количество колонок с фиксированной шириной колонки и отступа. Вместо этого для каждого куска страницы, в котором элементы должны быть выровнены по сетке, должна использоваться своя сетка с относительными размерами колонки и отступа. Количество столбцов в сетке может (и должно) меняться при изменении ширины экрана.

3. На одной странице и на одной ширине экрана может использоваться несколько разных сеток.

4. Рекомендуется ограничивать длину строки текста, так как при чтении слишком длинных строк взгляд все время теряет строку, когда возвращается в начало строки. Оптимальной является длина строки в 40-75 знаков, однако многие сайты ограничивают длину строки значением 120 знаков и даже более.

5. Как правило, отзывчивые сайты имеют гораздо более свободную компоновку, в ней больше «воздуха» - пустого пространства между элементами. Это облегчает восприятие информации. Такие сайты не перегружены, как многие сайты старого типа. В отзывчивых сайтах контент и

функционал распределены по вертикали. Кроме того, в отзывчивых сайтах принято разносить контент и функционал по отдельным страницам.

6. Не следует размещать в боковых панелях рекламные баннеры, акции, прогнозы погоды и т.п. На первом месте должен быть контент. Особое внимание стоит обратить на структуру внутренних страниц, а не только титульную.

Основные методы разработки отзывчивого дизайна

Рассмотрим это на примере системы управления контентом Joomla! Для того, чтобы создать такой дизайн необходимо отличные знания CSS и HTML. Одним из методов создания отзывчивого шаблона заключается в применении медиа запросов (HTML5). Данный метод также подходит для различных ориентаций экрана.

Другим методом создания отзывчивого дизайна является использование резинового макета CSS. Такие шаблоны хорошо смотрятся на всех используемых параметрах экранов. Метод строится на процентном выводе макета и объектов, которые располагаются на нем. Большая часть мобильных и современных браузеров правильно отображают страницы. Также необходимо выполнять его тестирование для различных браузеров.

Имеется уже множество шаблонов, которые используются для управления Joomla! на портативных устройствах, которые отличаются от ноутбуков и компьютеров. Одним из самых главных положительных моментов, которые помогут склониться к использованию отзывчивого дизайна, является удобство просмотра сайта с дизайном, который адаптируется под людей, имеющих проблемы со зрением.

Отзывчивый дизайн в системах управления контентом

Фреймворк для создания сайтов служит для поддержки разработки динамичных сайтов, веб-приложений и веб-сервисов.

Фреймворк (неологизм от англ. *framework* — каркас, структура) - программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Употребляется также слово «каркас», а некоторые авторы используют его в качестве основного, в том числе не базируясь вообще на англоязычном аналоге. Можно также говорить о каркасном подходе как о подходе к построению программ, где любая конфигурация программы строится из двух частей: первая, постоянная часть - каркас, не меняющийся от конфигурации к конфигурации и несущий в себе гнезда, в которых размещается вторая, переменная часть - сменные модули (или точки расширения)[2].

В настоящее время огромную популярность приобрели адаптивные и отзывчивые фреймворки.

Адаптивный фреймворк делает web-сайт дружелюбным для пользователя, какими бы устройствами он не пользовался. Они могут значительно ускорить процесс создания адаптивного веб-дизайна, а также

добавить много других опций на сайт, например, модальные окна, off-canvas меню и многое другое.

Компания Twitter совершила революцию в создании адаптивных и отзывчивых шаблонов для различных систем управления контентом. Основной идеей, которой руководствовалась компания была следующей: создать настраиваемый каркас, который можно применять в различных шаблонах. С другой стороны каждый такой шаблон должен поддерживать вызов необходимых функций фреймворка. Поэтому компании, разрабатывающие адаптивные/отзывчивые шаблоны параллельно разрабатывают и новые фреймворки.

На сегодняшний день существует 10 популярных фреймворков [3].

Foundadtion - это самый богатый опциями фреймворк, разработанный компанией Zurb. Доступна разнообразная библиотека самых разных опций, например, off-canvas меню, модальные окна, расширенная навигация canvas, ускорение графического процессора, fastclick.js и многие другие опции для разработчиков.

Bootstrap – это самый популярный адаптивный фреймворк, который применяется на миллионах сайтов. Многие темы на Themeforest имеют встроенный фреймворк Bootstrap. Лёгок в освоении, имеет дружелюбный интерфейс, отлично подходит для начинающих заниматься адаптивным веб-дизайном.

Gumby – это гибкий адаптивный фреймворк со множеством уникальных компонентов, которых нет в Bootstrap или Foundation. Gumby фреймворк разработан компанией Digital Surgeons. Сообщество у Bootstrap и Foundation намного шире, чем у Gumby. У фреймворка есть расширения, которые добавляют функциональность базовой установке.

Skeleton – это простой, легкий в использовании адаптивный CSS-фреймворк. Он не привязан к конкретному стилю и дизайну. С этим фреймворком можно быстро начать разработку адаптивного сайта без дополнительных функций. На сайте практически нет цветов. Это объясняется тем, что данный фреймворк задумывался как основа, некий каркас, который можно адаптировать под свой бренд и фирменные цвета/стили сайта.

Pure – это набор небольших адаптивных CSS модулей, которые можно использовать для любого веб-проекта. Это отличный инструмент, так как можно применять модули для любого проекта, даже если уже используется другой фреймворк или просто нужен только один модуль, а не весь фреймворк Pure.

The Responsive Grid System предлагает интересный подход к адаптивному дизайну, так как он может быть встроен в существующий HTML и CSS шаблон. Это очень простая адаптивная система в виде сетки, которую можно добавить к любому уже готовому сайту, который пока еще не имеет адаптивного дизайна.

Amazium – менее известный адаптивный фреймворк, который представляет собой простую сетку с намного меньшим количеством кодов, чем у других фреймворков.

Less Framework – очередная CSS система для создания адаптивного веб-сайта. Фреймворк содержит 4 верстки и три набора графики и все это основано на одной сетке. Преимущество Less Framework заключается в том, что его можно использовать для создания разных типов верстки и заменять элементы одной верстки на элементы другой верстки, меняя код.

Responsive – это один из самых легких HTML, JavaScript фреймворков из всех представленных. Имеет небольшой размер по сравнению с другими фреймворками.

Barekit – это еще один минималистический адаптивный фреймворк. Это совершенно новый адаптивный фреймворк, который был выпущен компанией AppendTo.

Использование фреймворков

Основным преимуществом использования фреймворка является LESS - динамический язык стилей, существенно расширяющий возможности CSS. С его помощью разработчики могут создавать переменные, вложенные колонки, управлять цветами и т.д. Так же LESS очень легко пользоваться - достаточно просто вставить код в страницы. Практически во всех фреймворках есть модуль вставки содержимого файла .less в каскадные таблицы стилей шаблона.

Основные инструменты фреймворка.

Сетки - заранее заданные размеры колонок, которые можно сразу же использовать.

Шаблоны - фиксированный или резиновый шаблон документа.

Типографика - описания шрифтов, определение некоторых классов для шрифтов таких как код, цитаты и т.п.

Медиа - представляет возможности управления изображениями и видео.

Таблицы - средства оформления таблиц, вплоть до добавления функциональности для обеспечения возможности сортировки.

Формы - классы для оформления не только форм, но и некоторых событий, происходящих с ними.

Навигация - классы оформления для вкладок, страниц, меню и панелей инструментов.

Алерты - оформление диалоговых окон, подсказок и всплывающих окон.

В большинстве фреймворков имеются дополнительные особенности и функции.

WYSIWYG - редактор для каркаса шаблона.

Мультиплатформенность - работа на всех операционных системах.

Присутствует **интерактивный просмотр и подсветка элементов**, но он работает если есть Google Chrome.

Быстрое редактирование, то есть можно редактировать файлы CSS и JavaScript прямо в файле HTML, что существенно экономит время.

Быстрый просмотр, благодаря нему можно просматривать цвета, которые используются в коде и изображения.

Быстрые подсказки, они показывают при подключении файла, стиля, скрипта или картинки не только путь, но и имя файла.

Фрейворки поддерживают несколько медиа-запросов в одном файле, что позволяет работать проекту на различных устройствах и с разными разрешениями экранов.

Отзывчивые и адаптивные шаблоны для CMS

Всё больше фирм, производящих шаблоны для таких популярных систем управления контентом (CMS) как Joomla! и WordPress склоняются к изготовлению адаптивных и отзывчивых шаблонов (WordPress называет их темами), основанных либо на фреймворке Twitter, либо создавая собственный фреймворк.

В таблице 1 приведены названия и ссылки фирм-производителей подобных шаблонов для CMS Joomla!

Таблица 1

Фирмы - разработчики адаптивных и отзывчивых шаблонов для CMS Joomla!

Фирма	URL-адрес	Фреймворк	Комментарий
GavickPro	https://www.gavick.com/	Gavern Framework	Кроме шаблонов разрабатывает компоненты и модули
JoomlaArt	https://www.joomlart.com/	T3 Framework	Шаблоны в основном платные. Из бесплатных самый популярный Purity III
JoomlaXTC	http://www.joomlaxtc.com/	XTC Framework	Распространяет шаблоны при членстве в клубе
Omega Theme	https://www.omegatheme.com/	Jarvis Framework	Распространяет шаблоны при членстве в клубе. Разрабатывает также темы для WordPress
Rocket Theme	http://www.rockettheme.com/	Gantry Framework	Разрабатывает также темы для WordPress, Magento, phpBB
Smart Addons	http://www.smartaddons.com/	YT Framework	Кроме шаблонов разрабатывает компоненты и модули. Шаблоны в основном платные
YOOtheme	https://yootheme.com/	Widgetkit	Шаблоны в основном платные.

Заключение

В результате того, что с помощью фреймворка можно реализовать адаптивные/отзывчивые сайты, снижаются расходы на создание дополнительных сайтов под другие мобильные технологии. Сайт, созданный с помощью адаптивных и отзывчивых шаблонов легче разрабатывать благодаря тому, что он бесплатный, снижается время на создание сайта, уменьшается количество людей, задействованных при создании сайта. Более того адаптивный/отзывчивый шаблон можно использовать неоднократно, изменяя

его с помощью соответствующего фреймворка, соответственно не надо тратить дополнительные средства для разработки нового сайта

Список литературы

1. Маркотт, И. Отзывчивый веб-дизайн: учеб. пособие / И. Маркотт - Москва: Манн, Иванов и Фербер, 2012. - 277 с.
2. Фреймворк // ru.wikipedia.org [Электронный ресурс] Режим доступа <https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA> - Дата доступа: 17.03.2016
3. 10 адаптивных фреймворков для веб-дизайна // rebill.me [Электронный ресурс] Режим доступа : <http://rebill.me/showthread.php?t=3560&highlight=%E0%E4%E0%EF%F2%E8%E2%ED%FB%F5+%F4%F0%E5%E9%EC%E2%EE%F0%EA%EE%E2+%E2%E5%E1-%E4%E8%E7%E0%E9%ED%E0> - Дата доступа: 15.02.2016

Гончаров Сергей Леонидович, старший преподаватель кафедры математического и информационного обеспечения экономических систем Учреждения Образования «Гродненский государственный университет имени Янки Купалы», sgonchar@grsu.by