

## Аспекты разработки адаптивных веб-интерфейсов

В. В. Захаркина  
Санкт-Петербургский государственный университет,  
Университет ИТМО,  
zakharkina@gmail.com

### Аннотация

Одним из существенных аспектов современной веб-разработки является анализ спектра актуальных аппаратных средств и, как следствие, пользовательского восприятия и проблем адекватного отображения веб-ресурсов на различных платформах.

Рассматриваются вопросы создания адаптивных веб-интерфейсов с акцентом на технологических предпосылках и возможностях реализаций.

**Ключевые слова:** веб-дизайн, веб-интерфейс, адаптивный веб-дизайн, CSS.

### Введение

Веб-дизайн с полным основанием можно рассматривать как междисциплинарную область веб-разработки, в которой неразрывно связаны технологические возможности, художественная стилистика, пользовательское восприятие.

Развитие аппаратного обеспечения, стандартов формального описания структуры и внешнего отображения веб-документов, программных возможностей не только делает более эффективным процесс веб-разработки, но и позволяет предложить пользователю новые функциональные возможности и сервисы, что требует от разработчиков новых интерфейсных решений. Знаменательно, что многие актуальные современные элементы интерфейса, по существу, являются переосмыслением ранних, ныне практически забытых.

Можно констатировать, что за последние годы в веб-дизайне сформировались определённые тенденции, которые можно выявить при анализе большинства вновь создаваемых ресурсов, а также старых сайтов, подвергнутых в последние годы полной функционально-визуальной переработке.

Представляется весьма полезным не только сформулировать эти тенденции, но и проследить истоки современных интерфейсных решений, а также отметить

технологические аспекты, способствовавшие их реализации. В статье рассматриваются вопросы, связанные с понятием адаптивного дизайна.

## **Адаптивный дизайн**

Понятие «отзывчивого веб-дизайна» [1] (*responsive web-design*) было предложено в 2010 году как концепция разработки веб-интерфейсов, ориентированная на адекватное отображение сайта на всех устройствах и всех платформах. В дальнейшем идея была развита, сделан акцент на необходимости комплексного учёта особенностей платформ и браузеров во всём многообразии их версий. В ряде англоязычных источников в этой связи при расширенной трактовке используется термин *adaptive web-design*. В русскоязычных же источниках, как правило, в любом контексте используется термин «адаптивный веб-дизайн», и мы будем следовать сложившейся традиции.

В определённом смысле, предпосылкой успеха концепции адаптивного дизайна у разработчиков и её поддержки в реализациях веб-ресурсов явилось беспрецедентно быстрое распространение мобильных устройств. Пользовательская аудитория оценила новые возможности доступа к ресурсам практически мгновенно, что заставило разработчиков искать новые решения.

Необходимость серьёзного переосмысления сложившихся ранее типовых подходов к проектированию веб-интерфейсов, несомненно, было продиктовано стремлением обеспечить комфортное и эффективное выполнение пользовательских задач. Как бы ни формулировали свои задачи держатели ресурсов, конечная их цель состоит в привлечении и дальнейшем удержании пользователей. Сайты коммерческой направленности нацелены на получение прибыли, некоммерческие же (общественные организации, профессиональные сообщества, инициативные проекты и т.д.) заинтересованы в информировании и привлечении к сотрудничеству. В любом случае, доступность ресурса для любых платформ и удобство его использования в типичных условиях становится важнейшей задачей разработчика.

Рассматривая концепцию адаптивного веб-дизайна в широком смысле, следует упомянуть о необходимости выработки кроссплатформенных и кроссбраузерных подходов по целому ряду вопросов, ещё не получивших окончательных решений. Это воспроизведение видео и аудио контента средствами браузера, проблемы представления интерактивных 3D объектов и сцен и т.д. Анализ этих проблем, характерных для ресурсов со сложным медиа-контентом, выходит за рамки текущей статьи и будет предметом дальнейших публикаций.

Далее рассматриваются существенные аспекты разработки веб-интерфейсов, актуальные для любых современных ресурсов.

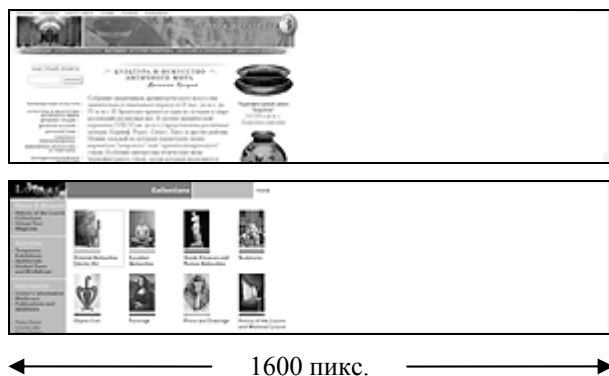
## **Проблемы адаптации макета веб-страницы к области просмотра**

Проблема адекватного отображения сайта на устройствах с различным разрешением экрана рассматривалась задолго до появления мобильных устройств. Многие разработчики середины — второй половины 90-ых (по крайней мере, в России) помнят эффект появления новых мониторов: это поначалу было событием. В первое время процесс представлялся

эволюционным и вполне контролируемым. Появление новых разрешений вполне могло быть сопровождено соответствующими дизайнерскими коррективами: сначала 600x480, потом 800x600, 1024x768...

Дальнейшее увеличение разрешения актуальных мониторов заставило дизайнеров ограничить ширину основного содержательного блока, дабы сохранить композиционное равновесие и обеспечить комфортное восприятие текста. К 2000 году, как правило, эта ширина составляла около 700–800 пикс., позже (и по сей день) 960–1200 пикс.

Ниже приведены фрагменты интерфейсов двух крупнейших музейных ресурсов от 2000 года (данные из архива <http://web.archive.org>), полученных при 100% масштаба браузера на типичном ноутбуке с горизонтальным разрешением 1600 пикс. Фрагменты обрезаны по высоте, рамка же позволяет оценить неэффективное использование пространства и — главное — неудобство восприятия. Разумеется, современные браузеры допускают масштабирование страницы, но при увеличении ухудшается качество изображений, нарушается композиция, излишне укрупняется текст.



**Рис. 1.** Фрагменты интерфейсов сайтов Государственного Эрмитажа и Музея Лувр. 2000 год.

Последовавшее бурное совершенствование мониторов и увеличение разрешения заставили дизайнеров искать решения, позволяющие эффективно использовать пространство, не нарушая целостность композиции.

Распространение же мобильных устройств вывело проблему в ряд первоочередных, поскольку вновь стали актуальными и очень малые разрешения. Настолько малые, что на протяжении ряда лет многие популярные ресурсы дублировали контент с целью создания содержательно зеркальных, но принципиально иначе оформленных «мобильных веб-версий». К настоящему же времени наблюдается тенденция к увеличению не только физических размеров, но и разрешения экранов смартфонов и аналогичных устройств.

Представление о текущем диапазоне разрешений даёт следующая таблица (источник — статистический ресурс <http://ru.screenresolution.org>).

**Таблица 1.** Фрагмент таблицы статистики разрешений мониторов (2016 г.).

№	Разрешение экрана монитора	%
1	1366x768 HD	18.44%
2	1920x1080 16:9 HD 1080	16.53%
3	1280x1024 5:4 SXGA	7.41%
4	1024x768 4:3 XGA	5.56%
5	1600x900 16:9 HD+ 900p	5.39%
6	360x640	4.93%
...		
15	320x568	1.26%
...		
2895	21120x3240	0.001%
2896	60300x10800	0.001%

Итак, диапазон актуальных разрешений заставляет задуматься о реализации **динамической подстройки** макета веб-страницы под текущее разрешение экрана и, более того, под размер области просмотра (ведь на большом мониторе пользователь вполне может уменьшить размеры окна браузера).

Динамическая подстройка макета веб-страницы под область просмотра предполагает целый спектр возможных решений. Это изменение размеров блоков и их взаимного расположения, скрытие либо вывод отдельных элементов, изменение параметров шрифта и т.д.

В принципе, реализация адаптивного макета может быть проведена с помощью программного сценария JavaScript (соответствующие решения предлагались в первой декаде 2000-ых на основе обработки события *resize*), однако этот подход представляется весьма громоздким и неэффективным.

Новые возможности открылись для разработчиков в связи с долгожданной поддержкой большинством браузеров основных возможностей стандарта CSS 3, особенно правил @media.

### **Предпосылки адаптивных решений**

Серьёзным шагом на пути к адаптивному дизайну стала возможность гибкого задания размеров элементов веб-страницы.

По существу, многие проблемы реализации дизайнерских идей были следствием назначения фиксированной ширины основному контейнеру содержимого (CSS свойство *width*). Действительно, предложить «идеальную» ширину для широкого диапазона разрешений в этом случае не представляется возможным. Незаполненное пространство на широких мониторах при уместном стилистическом оформлении не снижает эстетического впечатления и не создаёт дискомфорта при использовании ресурса. Но на небольших мониторах появляется горизонтальная прокрутка! Это абсолютно недопустимая ситуация, резко снижающая эффективность использования.

Примечательно, что ещё в CSS 2 (1998 г.) были определены свойства, позволяющие задать не только фиксированные значения ширины и высоты, но и максимальные/минимальные — то есть, диапазон. Наиболее полезным оказалось свойство *max-width* (максимальная ширина).

Задание ограничения на **максимальную ширину** контейнера не позволяет содержимому занять всё свободное пространство и тем самым обеспечивает должное композиционное решение. При просмотре же на устройствах с малым разрешением (либо при уменьшении размеров окна браузера) содержимое занимает 100% ширины окна, что, по крайней мере, обеспечивает комфортный просмотр.

Совместно со свойством *max-width*, как правило, используется процентное задание ширины вложенных блоков. Так, например, 50% обеспечивает расположение таких блоков по 2 в ряду, 25% — по 4 в ряду. Изменяя размеры окна, на абсолютном большинстве современных ресурсов можно наблюдать вполне «адаптивное» поведение элементов внутренней структуры.

Отдельно следует упомянуть о столь актуальных в последнее время «отзывчивых» (*responsive*) изображениях и слайдерах изображений. Таким элементам достаточно задать ширину 100% от ширины контейнера (*width: 100%*), а определить автоматически (*height: auto*). При изменении размеров контейнера в заданном диапазоне соответственно будет изменяться ширина изображения, пропорции же будут сохранены.

К сожалению, столь естественные и эффективные решения получили распространение лишь лет через 10 после представления стандарта CSS 2. С осторожностью назовём период с 2006 (выпуск MS Internet Explorer 7) по 2009 (MS IE 8).

Браузеры MS IE практически во всех своих версиях вызывают справедливые нарекания разработчиков, однако поддержка адекватного отображения сайта хотя бы для относительно свежих версий является обязательной. Поскольку пользовательская аудитория MS IE достаточно широка, выпуски версий, поддерживающих очередные возможности HTML и CSS, являются сигналом для разработчиков.

### Использование правил @media: CSS 3

Правила @media были введены в CSS 2 и обеспечивали возможность вариативного задания параметров отображения отдельных элементов веб-страницы при выводе на устройства различных типов (экран, принтер, синтезатор речи и т.д.). Эти возможности остаются актуальными и по сей день, например, при задании параметров вывода страницы на печать. Действительно, в таком случае вряд ли имеет смысл выводить декоративные элементы, навигацию, рекламные блоки и т.д. Скрытие их осуществляется соответствующими формальными объявлениями CSS.

Настоящим же прорывом в области реализации адаптивного дизайна стали расширения CSS 3, позволяющие определить десятки характеристик устройств вывода [2]. Это ширина, высота и пропорции как экрана устройства, так и области просмотра; разрешение экрана; цветовые параметры; ориентация (актуально для мобильных устройств) и т.д.

Любой, даже самый изысканный дизайнерский эскиз реализуется на основе описаний CSS, задающих для элементов документа цвета, размеры, взаимное расположение, поля, видимость или невидимость, особенности позиционирования и т.д. Возможность изменения этих параметров в зависимости от характеристик устройства просмотра средствами того же CSS и даёт желаемый эффект.

Приведём пример правила, обеспечивающего адаптивность интерфейса для следующей гипотетической ситуации. Представим себе горизонтальное главное меню, которое при малой ширине области просмотра (скажем, 720 пикс.) должно исчезать, уступая место характерной пиктограмме-«гамбургеру», активизация которой приводит к появлению «мини-меню».

Код HTML может выглядеть следующим образом:

```
<nav id="menu"> ... </div>           <!-- горизонтальное главное меню -->
<nav id="menu-mini"> ... </div>     <!-- пиктограмма -->
```

Возможный код CSS с правилом @media:

```
menu-mini { display: none; } /* по умолчанию пиктограмма скрыта */
...
@media all and (max-width: 720px) {
    menu { display: none; } /* главное меню скрывается */
    menu-mini { display: block; } /* пиктограмма показывается */
}
```

Разумеется, в нетривиальных случаях объявления CSS значительно объёмнее, ибо приходится учитывать особенности отображения целого ряда элементов. Ниже приведён пример простейшего, казалось бы, интерфейса шапки на сайте Государственного Эрмитажа (<https://www.hermitagemuseum.org>). Однако и здесь задействовано несколько элементов, изменяющих свою видимость и положение.

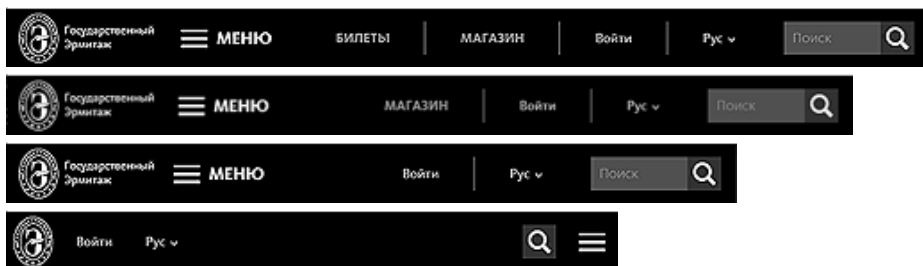


Рис. 2. Шапка сайта Государственного Эрмитажа при изменении ширины области просмотра.

Начало активного применения правил @media для реализации адаптивных макетов рискнём отнести к 2011 году, когда был выпущен MS Internet Explorer 9.

### **Адаптация элементов при прокрутке веб-страницы**

Реакцию элементов веб-интерфейса на изменение позиции прокрутки не относят к задачам адаптивного дизайна, и это представляется несправедливым.

Во-первых, значительная часть эффектов имеет не декоративное, а сугубо функциональное назначение. Продуманная стратегия обработки позиции прокрутки способна значительным образом улучшить пользовательское восприятие, повысить скорость работы, привлечь внимание к активным элементам веб-документа. Разумеется, повышение эффективности будет заметным для страниц с достаточно объёмным содержанием.

Во-вторых, с точки зрения программного сценария, настройка внешнего вида страницы в связи с изменением размеров области просмотра и модификация при прокрутке относятся к одному классу. А именно, представляют собой обработчики двух глобальных событий браузера: *resize* (изменение размера окна) и *scroll* (прокрутка).

Типичные решения при программной обработке позиции прокрутки: скрытие второстепенных элементов; появление элементов, способствующих эффективной навигации; фиксация существенных элементов относительно границ окна.

Предполагается, что пользователь инициирует прокрутку либо внимательно изучая содержимое (тогда следует скрыть вторичные элементы), либо – напротив – пытаясь сориентироваться на странице (при этом могут помочь дополнительные навигационные элементы). При прокрутке вниз, например, уместно сразу скрыть декоративные и малозначимые крупные элементы шапки. При этом основная навигация или поле ввода поискового запроса могут быть фиксированы у верхней границы окна. Как вариант, удачным решением может быть скрытие объёмного навигационного блока при прокрутке вниз и плавный его вывод при обратной прокрутке.

Весьма эффективной для длинных структурно сложных страниц является дополнительные навигационные элементы, сигнализирующие о текущем разделе и позволяющие перейти либо к выбранному другому, либо в начало страницы.

### **Эксперимент: беглый анализ музейных сайтов**

Для иллюстрации преимуществ адаптивного дизайна хотелось привести несколько позитивных примеров веб-ресурсов из области культуры и искусства. Решено было сделать небольшую подборку сайтов известнейших музеев. К удивлению автора, половина из них лишена признаков адаптивности. Более того, в формальных описаниях внешнего отображения элементов страниц используется фиксированное значение ширины контейнера. Как отмечалось выше, такой базовый подход не позволяет обеспечить дальнейшие гибкие решения. У адаптивных сайтов также обнаружили проблемы: некорректное

отображение в MS Internet Explorer 9. Несомненно, это устаревшая и неудачная версия, однако все типовые решения для неё вполне можно реализовать.

Добавив в подборку очень удачно (с технологической точки зрения) спроектированный сайт лондонского Музея естествознания, хотя он и выпадает из первоначальной категории «культура и искусство», получаем следующий результат.

Сайты, не содержащие адаптивных элементов:

- Русский музей (<http://rusmuseum.ru>);
- музей Капитолия (<http://www.museicapitolini.org>);
- Помпеи (<http://www.pompeisites.org>);
- Третьяковская галерея (<http://www.tretyakovgallery.ru>);
- музей Орсе (<http://www.musee-orsay.fr>);
- музей Лувр (<http://www.louvre.fr>).

Адаптивные сайты, некорректно отображающиеся в MS IE 9:

- музей Метрополитан (<http://www.metmuseum.org>);
- музей Ван Гога (<http://www.vangoghmuseum.nl/en>);
- государственные музеи Берлина (<http://www.smb.museum>).

Адаптивные сайты, поддерживающие MS IE 9:

- галерея Тейт (<http://www.tate.org.uk>);
- Государственный Эрмитаж (<http://hermitage.ru>);
- лондонский Музей Естествознания (<http://www.nhm.ac.uk>)

В чём причина несовременных технологических решений, трудно понять. Большинство рассмотренных сайтов подверглись редизайну всего несколько лет назад. Возможно, причина в том, что у музеев, помимо веб-ресурсов, есть и мобильные приложения. Действительно, пользователи мобильных устройств зачастую предпочитают загрузить специальное приложение. Однако, в отличие от прежних «мобильных веб-версий», мобильные приложения не являются «зеркалом» основного сайта и предоставляют неполное содержание.

Хотелось бы всё же надеяться на дальнейшие позитивные действия, направленные на комфортное воспроизведение на широком спектре устройств.

## Заключение

Реализация эффективных интерфейсов остаётся одной из ключевых задач веб-разработки. Серьёзным стимулом к формулированию новых задач является развитие аппаратных средств и усложнение функциональности, предлагаемой веб-сервисами. При этом совершенствование формальных языков и средств веб-разработки зачастую опережает аппаратный прогресс.

Основным вызовом для разработчиков является запаздывание поддержки новых стандартов рядом платформ и браузеров. Удачные и востребованные решения зачастую годами остаются на уровне концептов. Тем не менее, процесс создания адаптивных веб-интерфейсов вышел из кризиса и в дальнейшем имеет хорошие перспективы.



## **Литература**

- [1] Ethan Marcotte. Responsive Web Design. May 25, 2010. URL: <http://alistapart.com/article/responsive-web-design> (дата обращения: 12.05.2016).
- [2] CSS3 @media Rule. URL: [http://www.w3schools.com/cssref/css3\\_pr\\_mediaquery.asp](http://www.w3schools.com/cssref/css3_pr_mediaquery.asp) (дата обращения: 12.05.2016).

## **Aspects of Responsive Web-Interfaces Development**

V.V. Zakharkina  
St.Petersburg State University, ITMO University

Problem of appropriate display of web resources on various platforms and devices is the essential aspect of modern web development. Issues of creation of responsive web-interfaces are considered with the focus on technological pre-requisites and implementation opportunities.

Keywords: web-design, web-interface, responsive web-design, CSS.