**Сведения об авторе:**

**ФИО:** Кузменков Анатолий Сергеевич

**Академическое звание:** Магистр технических наук

**Наименование учебного заведения:** Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики (Университет ИТМО)

**Адрес электронной почты:** njkz-09@yandex.by

**Контактный телефон:** +375298526443

**Рубрика:** Технические науки

**Количество журналов:** 0

**Компонентный подход разработки приложений во front-end.**

За несколько последних десятилетий человечество проделало огромный путь в области компьютерных технологий в целом, и веб-технологиий в частности. За эти годы многократно увеличилась производительная мощность техники, а так же пропускная способность каналов связи. Веб-технологии затронули большинство сфер деятельности современного общества. Так веб-технологии получили широкое распространение в сферах электронного бизнеса, средствах массовой информации, литература, музыка, кино, связь и общение.

Однако при все нарастающих темпах использования всемирной паутины, появлялись новые требования от разработчиков в плане оптимизации процесса разработки и требования к производительности. Не смотря на развитие новых подходов и технологий основополагающими во front-end по прежнему являются *HTML*, *CSS* и *JavaScript*.

В плане организации работы современные реалии заключается в необходимости разработки приложений в рамках команд разбросанных по всему миру. Такая ситуация приводит к необходимости создания приложений с максимально простым для понимания синтаксисом, а также требует совершенствования инструментов разработчика.

И так современные реалии приводят к необходимости организации всего приложения или больших частей в виде универсальных модулей и компонент с максимально понятным синтаксисом. И дальнейшее их использование в большом количестве проектов.

Таким образом большинство современных технологий в той или иной степени используют компонентный подход.

С появления первых сетей для обмена информации между американскими университетами началось совершенствование методов доступа и распространения информации. Постепенно происходил переход от простого набора статических документов, оформленных с помощью специальных языков разметки (*HTML*, *XML*), произошёл переход к сложным веб приложениям. Кроме того изменились сами цели использования веб-технологий. В настоящее время больше не достаточно предоставлять информацию с простейшим графическим оформлением. Современные реалии требуют сложную и эффективную обработку данных.

С начала развития сетей *internet* постоянно увеличивалось их использование во всех сферах жизнедеятельности человека. Однако темп совершенствования оборудования не позволяет обеспечивать все увеличивающиеся потребности. В какой-то момент стало очевидным необходимость создание мощного средства программирования на стороне клиента, позволяющего не только переложить часть нагрузки на клиента, но и создавать более эффективные веб-приложения. Наиболее удачным решением стало создание и внедрение скриптового языка программирования *JavaScript* [1, с. 11]. В настоящее время данный язык используется в большинстве разработанных и разрабатываемых веб-приложений.

В настоящее время *JavaScript* и широко используются в следующих областях:

1. Электронный бизнес. По результатам исследований, большинство ресурсов Интернета так или иначе связаны с коммерческой деятельностью. Интернет используется для рекламы и непосредственной продажи товаров и услуг, для маркетинговых исследований, электронных платежей и управления банковскими счетами.

2. Средства массовой информации. По жанрам интернет-издания не отличаются от офлайновых – есть новостные сайты, литературные, научно-популярные, детские, женские и т. п. Однако, если офлайновые издания выпускаются периодически (раз в день, неделю, месяц), то интернет-издания обновляются по мере появления нового материала. Также существует интернет-радио и интернет-телевидение.

3. Литература, музыка, кино. Электронные библиотеки, доступные через Интернет, содержат огромное число произведений. При этом многие книги, доступные в сети, давно стали библиографической редкостью, а некоторые и вообще не издавались.

4. Связь. Электронная почта в настоящее время является одним из наиболее используемых средств связи. Также популярны IP-телефония и использование таких программ, как *Skype*. В последние годы популярность обрели мессенджеры, передающие сообщения через Интернет, они стали вытеснять из повседневной жизни людей сотовую связь, которая в сравнении с ними чаще всего уступает в функциональности, скорости, а также стоимости.

5. Общение. Интернет является способом массового общения людей, объединённых различными интересами. Для этого используются интернет-форумы, блоги и социальные сети. Социальные сети стали своего рода интернет-пристанищем, где каждый может найти техническую и социальную базу для создания своего виртуального «Я». При этом каждый пользователь получил возможность не просто общаться и творить, но и делиться плодами своего творчества с многомиллионной аудиторией той или иной социальной сети.

6. Краудсорсинг. Интернет оказался хорошим средством для решения общественно значимых задач силами множества добровольцев, координирующих при этом свою деятельность.

Не смотря на постоянное развитие и совершенствование, *JavaScript* столкнулся с одной сложностью. Разработчики браузеров на протяжении длительного времени не могли договориться об единообразии работы с веб-документами, что требовало разработки отдельного варианта скрипта для каждого браузера. Существует два распространённых способа решения этой проблемы:

1. Первый путь заключается в использование полифиллов. Полифилл (англ. *polyfill*) – это библиотека, которая добавляет в старые браузеры поддержку возможностей, которые в современных браузерах являются встроенными [2]. Например библиотека ES5 shim позволяет использовать функции добавленные стандартом *ES5* в *Internet Explorer8*.

2. Использование *JS*-фреймворков – позволяет достичь высокого уровня абстракции и обеспечить кросс-браузерность приложения за счет использования готовых решений. Среди наиболее известных можно отметить *Ember.js, Adobe Spry, AngularJS, Dojo Toolkit, Ext js, jQuery*, и др.

Преимущества использования JavaScript фреймворков [3]:

1. Эффективность – проекты, которые раньше заняли бы месяцы и сотни строчек кода, сейчас могут быть реализованы намного быстрее с хорошо структурированными готовыми шаблонами и функциями.

2. Безопасность – лучшие *JavaScript* фреймворки имеют фирменную систему безопасности и поддерживаются крупным сообществом, члены которого и просто пользователи выступают в роли тестировщиков.

3. Расходы – большинство фреймворков с открытым кодом и бесплатны. Поскольку они помогают программистам быстрее разрабатывать пользовательские решения, итоговая цена веб приложения будет ниже.

Рассмотрим более подробно возможности *JS*-фреймворков их достоинства и недостатки на примере *jQuery* и *Ext JS*.

Библиотека *jQuery* содержит функциональность, полезную для максимально широкого круга задач. Тем не менее, разработчиками библиотеки не ставилась задача совмещения функций, которые подошли бы всюду, поскольку это привело бы к большому коду, большая часть которого не востребована. Поэтому была реализована архитектура компактного универсального ядра библиотеки и плагинов. Это позволяет собрать для ресурса именно ту JavaScript-функциональность, которая на нём была бы востребована [4 c. 247].

Таким образом достоинства использования jQuery:

1. Облегчает манипулирование моделью документа DOM;

2. Множество различных эфектов;

3. Выполнение Ajax-запросов;

4. Кросс-браузерная совместимость;

5. Модульность jQuery, Вы можете нарастить функционал, подключая модули;

6. Простота использования.

Недостатки:

1. Скорость выполнения. Да, чистый JavaScript работает быстрее (в умелых руках), но говорить о быстродействии не совсем правильно (растет производительность пользовательских компьютеров, да и сами разработчики постоянно наращивают быстродействие самого фреймвёрка).

*Ext JS* представляет собой *JS*-фреймворк для построения интерактивных веб-приложений с использованием *AJAX*, *DHTML* и *DOM*. Библиотека включает в себя широкий набор элементов управления для использования в веб-приложениях: текстовые поля, дата и время, числовые поля, списки, элементы выбора, редактор HTML, деревья и прочее. Многие из этих элементов позволяют легко настроить взаимодействие с сервером через AJAX запросы.

Достоинства [5 ст. 21]:

1. большой набор виджетов позволяет в короткие темпы разработать высоко функциональный пользовательский интерфейс;

2. приближенность пользовательского интерфейса к оконному приложению;

3. оптимизация интерфейса для мобильных устройств;

4. организация эффективной работы с большими объемами данных;

5. наличие проработанной документации в том числе и на русском языке.

Недостатки библиотеки заключены в ограниченных возможностях стилевого оформления приложения, а так же в больших размерах самой библиотеки, что делает её не эффективной при создании малых проектов. Так же не мало важным недостатком является лицензия, она позволяет использовать библиотеку только на не коммерческих проектах [5 ст 22].

Развитие всевозможных фреймворков избавляет программистов от необходимости решения типовых однообразных задач, а так же позволяет использовать многочисленные наработки, не заботясь об совместимости на различных браузерах.

Однако следует внимательно относиться к выборы фреймворка, дабы максимально использовать его возможности. На примере рассмотренных *jQuery*, *Ext JS* можно сделать вывод, что каждый фреймворк не является универсальным. Задачи эффективно решаемые одной библиотекой не будут столь эффективны при решении с испльзованием иной. Таким образом достоинством любого JS-фреймворка является наличие механизмов для реше-ния конкретного спектра задач. Так jQuery позволяет в короткие сроки разрабатывать такие сайты-визитки, информационные и новостные ресурсы. При этом нет необходимости заботиться о крос-браузерности полученных сайтов. Использование *Ext JS* будет более эффективным на крупных проектах связанных с обработкой и анализом больших объемов данных.

**Библиографический список:**

1. Lindley, C. JavaScript Enlightenment. O’Reilly Media, 2012. 166 p.

2. JavaScript.ru. Полифилы. URL: https://learn.javascript.ru/dom-polyfill (Дата доступа: 30.05.2017).

3. DailyJS - JavaScript Blog. Building A JavaScript Framework. URL: http://dailyjs.com (Дата доступа: 30.05.2017)

4. Бенедетти, Р., Крэнли Р. Изучаем работу с jQuery. Спб.: Питер, 2012. 508 с.

5. Garcia, J. Ext JS in Action. Manning Publication, 2011. 495 p.