Предисловие

Как создатель JavaScript, я бы хотел сказать несколько слов о том, что такое JavaScript, как появился этот язык и для чего нужен. Книга, которую вы держите в руках, поможет вам освоить новые вершины программирования.

JavaScript — это плод неуемного желания предоставить разработчикам HTML-документов средства динамического управления всеми его объектами, которые описывались бы исключительно в коде Web-страницы. Теперь такой подход кажется простым и очевидным, но весной 1995 года эта идея была весьма нестандартная. Она была связанна с серьезными несоответствиями между принятыми прописными истинами (принципы HTML предписывают создавать только документы со статической структурой) и принципами динамического изменения документов (речь идет об аплетах Java, признанных единственным реальным способом "оживить" и расширить возможности Web-страницы). Раз уж пришлось затронуть подобные вопросы развития среды Internet, то рассмотрим детальнее особенности языка JavaScript.

- В первую очередь, он имеет облегченный синтаксис языка Java. Синтаксис "натурального языка" HyperTalk представлялся мне вполне приемлемым. Но после того, как один мой хороший знакомый одолжил мне книгу *The Complete HyperCard Handbook*, написанную Дэнни Гудманом, я все больше стал склоняться к мысли об использовании в качестве базиса для нового языка именно Java, особенно в свете бурного его развития. Если разрабатываемый язык синтаксически будет подобен Java, то с точки зрения тех программистов, которые переориентируются с одного языка на другой, такое совпадение будет только приветствоваться. Однако обязательность принципа декларирования классов и типов Java или использования точки с запятой после каждого оператора в месте окончания строки не вызвала у меня должного восторга. Ведь создание сценариев для большинства пользователей сродни написанию коротких отрывков кода — оно должно выполняться быстро и четко.
- События для HTML-элементов. Кнопки должны управляться обработчиками события `onClick`. Документы загружаются в окна и выгружаются из них, так что в сценарии должны быть представлены еще и обработчики события `onLoad` и `onUnload`. Пользователи в сценарии используют различные формы — отсюда вытекает необходимость в обработчике события `onSubmit`. Хотя в принципе и не обходя по гибкости систему сообщений HyperCard (обработчики последнего ориентированы на использование системы обозначений типа `onEvent`), события в JavaScript позволяют разработчикам HTML-документов взаимодействовать с пользователями с помощью удаленных серверов и быстро реагировать на их действия и операции, выполняемые браузером. С использованием рекомендаций по обработке событий W3C DOM 2 в современных браузерах удалось посредством JavaScript полностью реализовать гибкий контроль над событиями.
- Объекты без классов. Язык программирования Self оправдал идею использования основанного на прототипах принципа наследования. Для JavaScript я хотел использовать один прототип для одного объекта (для большей простоты и надежности), который основывается по умолчанию на функции, вызываемой с помощью оператора `new` (для

согласованности с Java). Чтобы обойтись без характерных для такой ситуации методов и функций, все функции задают объекты, вызывающие свойства с помощью ключевого слова `this`. Хотя свойства активно не использовались вплоть до версии NN3, их прототип уже был заложен во второй версии в виде заключаемого в кавычки текста, который интерпретировался как объект (прототип объекта `String`, к которому пользователь мог присоединять методы).

- Генерирование HTML-кода. Внедрение JavaScript в HTML способствовало развитию следующей идеи. А что, если HTML-код будет генерироваться в результате загрузки обычного текста (задаваемого в сценарии) в ходе его выполнения? Возможности этого подхода вышли далеко за пределы простого управления датой или временем внесения последних изменений. Эта простая идея позволила создавать сложные программные конструкции, позволяющие отслеживать данные многоуровневой структуры таблиц, где все повторяемые элементы задаются в цикле сценария, в то время как вся информация, заносимая в таблицы, обрабатывается в JavaScript минимальным набором операторов и сохраняется в файлах каталогов или мини-базах данных.

Вначале я полагал, что JavaScript найдет себе применение, прежде всего, в качестве средства ввода данных на HTML-формах. Но через некоторое время был удивлен, узнав, сколько дизайнеров, работающих с Web, используют его в своих проектах и применяют готовые JavaScript-приложения в HTML-документах для решения самых разнообразных задач. Web-дизайнеры давно искали способы быстрого и эффективного создания серьезных приложений с помощью минимального набора простых операторов. О популярности языка можно было судить по ответной реакции пользователей. В конечном счете разработчики (а также пользователи) настояли на реализации браузерной поддержки спецификации, известной под названием *Dynamic HTML* (презабавнейшим образом она описана по адресу: `http://www.javascript-games.org/`).

В то время как армия Web-разработчиков примерялась к возможностям JavaScript, стало явным несомненное преимущество среды создания сценариев над старым программным достоянием. Это относится не только к большой относительной простоте языков HTML и JavaScript, но также и к минимальным требованиям к разработчику Web-страниц. Ему не нужно обладать опытом программирования, чтобы управлять изображениями, обрабатывать события и создавать красочные эффекты, что невозможно при создании больших приложений с помощью громоздких языков, программируемых традиционным способом.

Превосходство JavaScript в Web сегодня подтверждает нашу безоглядную веру в ценность языка создания сценариев для разработчиков HTML-документов. Удерживая на низком уровне планку "требований", среда HTML (вместе с JavaScript) сделала дизайнерами миллионы обычных граждан. Позволяя "обрабатывать события", JavaScript помог тысячам дизайнеров стать настоящими программистами. Гарантом подобности Web-программирования и разработки сложных приложений является браузер Mozilla, в котором все элементы графического интерфейса и даже некоторые специальные элементы, а также программные модули полностью управляются средствами JavaScript, каскадных таблиц стилей (CSS), специальных языков, созданных на основе XML.

JavaScript является общим языком, применимым даже вне среды HTML и XML. Он поддерживается многими серверами, инструментальными средствами автоматизации, надстройками браузеров, а также браузерами других типов (это касается даже таких, казалось бы, "отстраненных" концепций как, например, управление трехмерным текстом). Соответствующим международным стандартом является ECMA-262 (ISO 16262), он уже доработан до третьего издания. Но, по сравнению с такими языками, как Perl или даже Java, он все еще относительно молодой. Работа в рамках технического комитета ECMA над четвертой версией, в которой должны поддерживаться типы, классы, а также средства организации разработки новых поколений программных

средств, идет полным ходом (посмотрите документацию в разделе JS2 на узле технической поддержки комитета ECMA по адресу `http://www.mozilla.org/js/language/js20/`).

Совершенно очевидно, по крайней мере для меня, что JavaScript не удержался бы на плаву без продуктивной, лояльной и терпеливой поддержки консорциума разработчиков. Должен выразить им свою искреннюю признательность. Те из них, кто тестировал бета-версию Netscape Navigator 2, распространял в виде почтовых сообщений и сообщений групп новостей жизненно важные доработки и затрагивал ключевые вопросы, по праву могут считаться крестными родителями языка. Поддержка разработчиков и постоянная обратная связь с конечными пользователями способствует заслуженному успеху JavaScript.

Книга, которую читатель держит в своих руках, является результатом тех "бессонных ночей", через которые прошли разработчики языка, опытные эксперты и преподаватели. Дэнни не мог и предположить, какое впечатление произведет на меня его книга о *HyperCard*. Именно она была моим настольным пособием в процессе создания JavaScript в 1995 году. Его энергия, участие и легкость изложения материала помогли мне удержаться в русле создания языка "для всех". Поэтому чрезвычайно приятно писать предисловие к пятому изданию этой книги, которая стоит проведенных им "бессонных ночей".

Настоятельно рекомендую всем, кто хочет освоить JavaScript, прочитать книгу Дэнни Гудмана *JavaScript. Библия пользователя*. Особенно это относится к тем, кто работает с HTML и успел "набросать" несколько сценариев или программ, — у вас есть уникальная возможность пройти по тернистой дороге создания сценариев вместе с опытным проводником и учителем.

Брендан Эйх

The Mozilla Organization (`http://www.mozilla.org`)