

# [Обучение Ext JS] :: Часть 1. Основы основ, или с чем едят Ext JS

Фев 17 2010Использование библиотеки Ext

После того как мы добавили **Ext** на страницу, можно начать писать код, который будет ее использовать. В первом примере используем Ext для отображения диалогового окна сообщения. Звучит не очень, но это всего-лишь начало!

## Время действовать

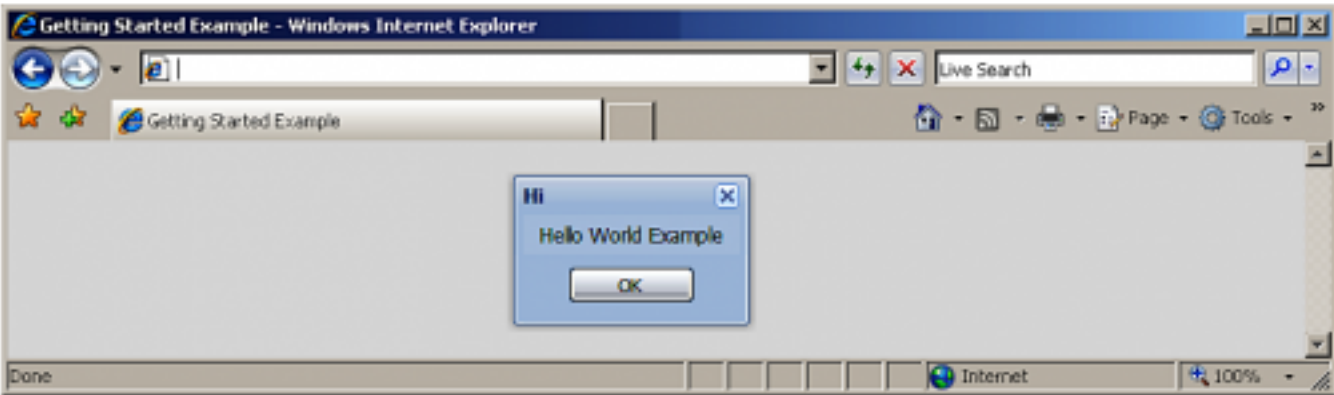
Мы можем запустить код **Ext** добавив в заголовок страницы часть скрипта. Сразу после того места, где мы прописывали библиотеку. Наш пример будет вызывать предупреждающее диалоговое окно в стиле **Ext**:

[Исходный код примера:](#)

## Исходный код примера:

```
<html>
<head>
  <title>Getting Started Example</title>
  <link rel="stylesheet" type="text/css"
    href="lib/extjs/resources/css/ext-all.css" />
  <script src="lib/extjs/adaptor/ext/ext-base.js"></script>
  <script src="lib/extjs/ext-all-debug.js"></script>
  <script>
    Ext.onReady(function(){
      Ext.Msg.alert('Hello', 'World');
    });
  </script>
</head>
<body>
  <!-- Nothing in the body -->
</body>
</html>
```

Мы не будем пока вдаваться в подробности, и выяснять что именно делает наш код. Сначала проверим правильность установки библиотеки Ext. Открыв нашу страницу в браузере, мы должны увидеть примерно такое оповещение:



Выглядит как "настоящее" диалоговое окно. Вы даже можете перетаскивать его куда угодно (разумеется в пределах страницы). Это потому, что оно и не окно вовсе, а набор тегов DIV и картинок собранных вместе для имитации диалога. Так же присутствующие там кнопки **Close** и **Ok** подсвечиваются при наведении курсора. Неплохо для одной строчки кода. **Ext** делает почти всю работу за нас, а по ходу книги мы научимся заставлять делать его еще больше!

**ВАЖНО:** Вы должно быть заметили, что мы работаем с пустым документом, у которого в теле нет элементов. Ext не требует предварительной разметки. Он создает все необходимое самостоятельно

## Пример

Давайте рассмотрим пример кода, приведенного выше повнимательнее. Каждый используемый нами компонент Ext начинается с **"Ext"** и с большой долей вероятности заключен в функцию *onReady* которую мы детально рассмотрим в следующей главе.

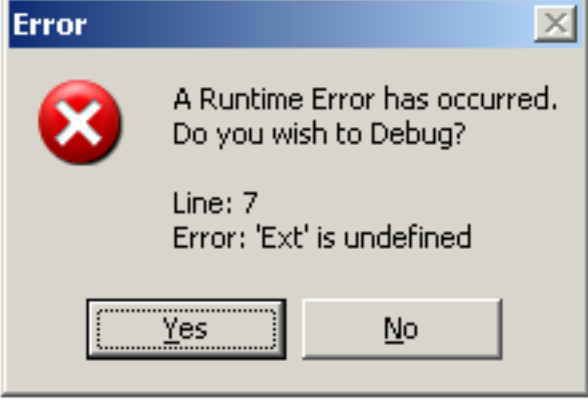
```
Ext.onReady(function(){
  Ext.Msg.alert('Hello', 'World');
});
```

У **Ext** очень удобный интерфейс. Код можно уже прочитать как предложение —Когда **Ext** будет готов, отобразить окно сообщения с Hello в качестве заголовка и World как тела.

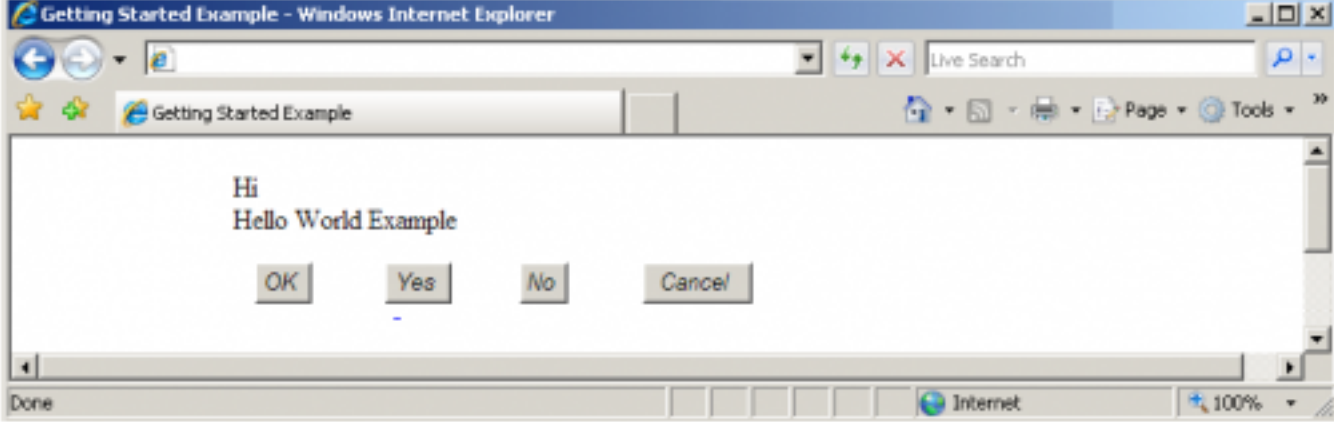
Наше окно начинается с Ext.Msg, которая является отправной точкой для всех окон такого типа и сокращением для "MessageBox". А "alert" указывает **Ext** какой стиль использовать.

## Не работает?

Если библиотека была установлена неправильно, то у нас появится окно с ошибкой:



Это сообщение значит, что библиотека **Ext** не была загружена. Обычно, это происходит если неправильно указать путь к одному из файлов. Перепроверьте пути и убедитесь что файлы существуют. Если же все указано правильно, вы должны увидеть папку adaptor рядом с файлами *ext-all.js* и *ext-all-debug.js* в вашей папке *lib/extjs*.



Если такое случилось убедиться в том, что папка *resources* была извлечена из **SDK**, и вы указали правильные пути. Эта папка должна находиться в *lib/extjs*.

## Адаптеры

Когда **Ext** только разрабатывалась (и называлась "yui-ext"), ей требовалась библиотека YUI для выполнения фоновой работы. Позже, была добавлена возможность использовать две другие структуры — jQuery или Prototype с Scriptaculous (Protaculous).

Это удобно в тех случаях, если мы уже используем другие библиотеки, или же нам кажется что другая библиотека может справиться с поставленной задачей лучше, чем поставляемые по умолчанию. И мы можем заставить их работать вместе с **Ext**, используя правильный адаптер. В любом случае **Ext** будет вести себя совершенно так же, а все ее компоненты будут работать одинаково вне зависимости от того, какой адаптер вы выберете.

Также в **Ext** есть свой собственный адаптер. Если вам не понравилась (или не подошла) ни одна из сторонних библиотек или структур используйте его.

## Использование адаптеров

Для их использования, вы должны сначала прописать ту внешнюю библиотеку, которую хотите использовать, а затем прописать зависимый файл адаптера, расположенный в папке adaptor **SDK Ext**. Пример нашего кода использует встроенный адаптер **Ext**. Для использования библиотеки замените строку, стоящего по умолчанию скрипт адаптера **Ext** на ту, библиотеку которого вы хотите использовать:

**Ext** адаптер по умолчанию:

```
<script src="lib/extjs/adaptor/ext/ext-base.js"></script>
```

Для **jQuery** пропишите эти файлы в заголовке документа:

```
<script src="lib/jquery.js"></script>
<script src="lib/jquery-plugins.js"></script>
<script src="lib/extjs/adaptor/jquery/ext-jquery-adaptor.js">
  </script>
```

Для **YUI** пропишите эти файлы в заголовке. Файл *utilities* находится в папке *build/utilities* загрузки библиотеки YUI:

```
<script src="lib/utilities.js"></script>
<script src="lib/extjs/adaptor/yui/ext-yui-adaptor.js"></script>
```

Для **"Prototype + Scriptaculous"** пропишите эти файлы в заголовке:

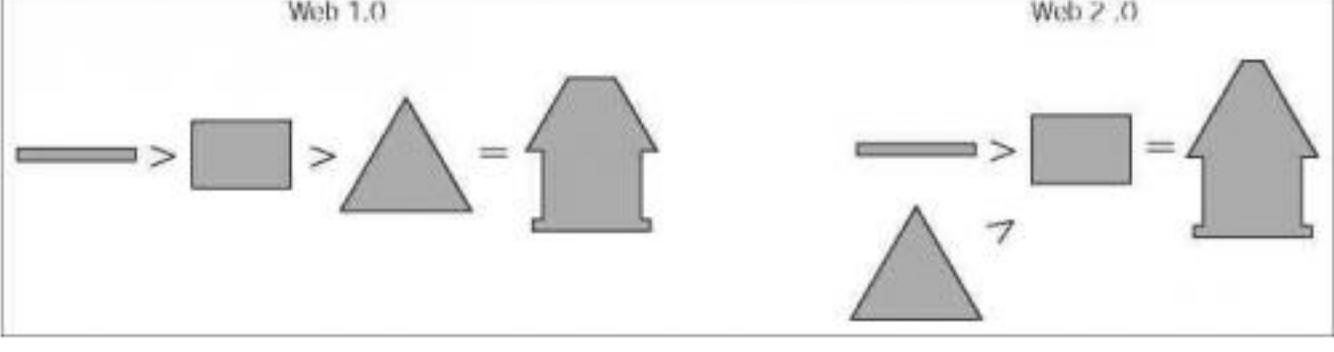
```
<script src="lib/prototype.js"></script>
<script src="lib/scriptaculous.js?load=effects"></script>
<script src="lib/extjs/adaptor/prototype/
  ext-prototype-adaptor.js"></script>
```

После того как адаптер и основные библиотеки будут прописаны, нам необходимо прописать файл *ext-all.js* или *ext-all-debug.js*.

## Я несинхронный!

Web 1.0 выполняет код строка за строкой, дожидаясь пока закончиться выполнение одного элемента код и лишь потом переходя к следующему. Очень похоже на то, как строят дом. Прежде чем строить стены надо заложить фундамент, и только потом уже можно переходить к крыше.

При помощи **Ext**, мы можем начать строить с крыши, даже если фундамент только еще находится в стадии разработки! Представьте, что крыша нашего дома собирается на фабрике, и в то время пока мы строим фундамент, затем стены, а потом мы заканчиваем строительство, устанавливая уже готовую крышу.



Благодаря тому что крыша, теперь не зависит от фундамента, нам не нужно придерживаться строчного подхода в веб-разработке. Ext помогает нам, давая события и операторы, к которым мы можем прикрепить функционал. Мы можем настроить событие, которое будет ждать до тех пор, пока не будут закончены стены, а затем само поставит крышу на место.

Такой способ может быть непривычен для тех, кто вырос в веб-программировании. Но уверяем, вы очень быстро освоите его.

**ВАЖНО:** Обычные предупреждения в **JavaScript** приостанавливают выполнение кода, что может привести к самым неожиданным результатам. Не используйте встроенные предупреждения **JavaScript**. Вместо них воспользуйтесь приложением Ext's *MessageBox*, которое не приостанавливает выполнение кода.