[Обучение Ext JS] :: Часть 2. Основы Ext

Фев 17 2010**Разжигание огня**
Теперб мы можем начать задать какую-нибудь реакцию нашей страницы в ответ на действия пользователя. Мы добавим оператор-переключатель, уторый позаботится о нажатии на кнопку Yes. Функция строки ввода займется третьим параметром, который является исполняемой функцией, и бует выполнен после нажатия кнопки Yes. Мы определяем это таким образом, чтобы функция проверила число введенное в диалоге ввода и если оно равно офису, то выводится в DIV на нашей странице. Если же нет, то выводиться будет стоящий по умолчанию текст Dull Work. Код так же применит тот же самый стиль и к DIV, использующее в качестве фона изображение степлера "Swingline".

Исходный код примера:

Исходный код примера:

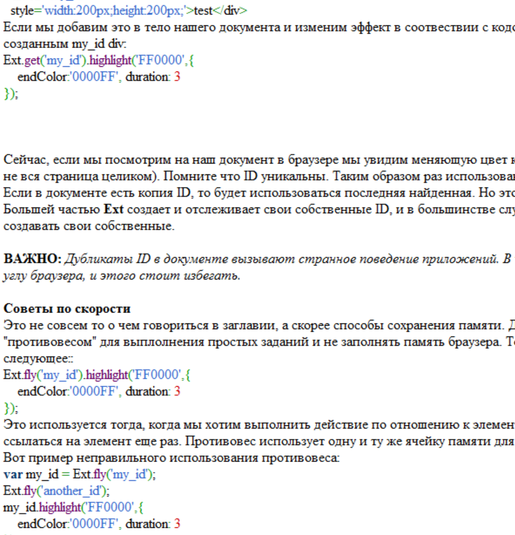
```
case 'yes':
  Ext.Msg.prompt('Milton', 'Where is it?', function(btn,txt)
{
      if (txt.toLowerCase() == 'the office') {
        Ext.get('my_id').dom.innerHTML = 'Dull Work';
      }else{
        Ext.get('my_id').dom.innerHTML = txt;
      }
      Ext.DomHelper.applyStyles('my_id',{
        background: 'transparent
          url(images/stapler.png) 50% 50% no-repeat'
    });
  });
break;
```

В случае если нажимается кнопка No, документ оформляется следующим образом:

Исходный код примера:

Исходный код примера:

```
case 'no':
  Ext.Msg.alert('Milton',
    'Im going to burn the building down!',
    function() {
      Ext.DomHelper.applyStyles('my_id',{
        'background': 'transparent
          url(images/fire.png) 0 100% repeat-x'
      });
      Ext.DomHelper.applyStyles(Ext.getBody(),{
        'background-color': '#FF0000'
      });
      Ext.getBody().highlight('FFCC00',{
        endColor:'FF0000',
        duration: 6
      });
  });
break;
```



Рабочая лошадка—Ext.get
Ext хорошо работает потому. что у него есть основание предоставляющие доступ к **DOM**, и большому количеству функций, которые позволяют управлять **DOM**. Из этих функций get одна из самых часто используемых.
Ext.get('my_id');
Это даст нам доступ к элементам документа с ID, my_id. Если мы посмотрим на первый пример, то заметим что он использует getBody для того чтобы извлекать элементы тела и применяет к нему наш эффект. Давайте, вместо этого, переключимся на использование my_id. Но сначала нам надо создать элемент my_id в нашем документе:

```
<div id='my_id'
  style='width:200px;height:200px;'>test</div>
```

Если мы добавим это в тело нашего документа и изменим эффект в соотвествии с кодом, а не телом, то тогда эффект будет только с моим созданным my_id div:

```
Ext.get('my_id').highlight('FF0000',{
    endColor:'0000FF', duration: 3
});
```

Сейчас, если мы посмотрим на наш документ в браузере мы увидим меняющую цвет квадратную коробку со стороной в 200 пикселей (а не вся страница целиком). Помните что ID уникальны. Таким образом раз использовав my_id, мы не можем использовать его повторно. Если в документе есть копия ID, то будет использоваться последняя найденная. Но это считается ошибкой, а не способом дизайна. Большей частью **Ext** создает и отслеживает свои собственные ID, и в большинстве случаев мы будем находить и использовать их, а не создавать свои собственные.

ВАЖНО: *Дубликаты ID в документе вызывают странное поведение приложений. В частности их окна возникают в верхнем левом углу браузера, и этого стоит избегать.*

Советы по скорости
Это не совсем то о чем говориться в заглавии, а скорее способы сохранения памяти. Для этого используют то, что называют "противовесом" для выплолнения простых заданий и не заполнять память браузера. Тот же самый эффект мы получим добавив в код следующее::

```
Ext.fly('my_id').highlight('FF0000',{
    endColor:'0000FF', duration: 3
});
```

Это используется тогда, когда мы хотим выполнить действие по отношению к элементу в одной строке кода и нам не надо будет ссылаться на элемент еще раз. Противовес использует одну и ту же ячейку памяти для одного и того же задания.
Вот пример неправильного использования противовеса:

```
var my_id = Ext.fly('my_id');
Ext.fly('another_id');
my_id.highlight('FF0000',{
    endColor:'0000FF', duration: 3
});
```

Поскольку противовес использует одну и ту же ячейку памяти, то к моменту когда м запускаем функцию по ссылке my_id, память уже изменила ссылку на another_id.

Заключение
Используя несколько строчек кодамы создали веселую программу, кторая будет развлекать васчасами. Ну, может не часами, но уж несколько минту уж точно. Тем не менее затронули только начало основного функционала и пользовательского интерфейса обычного десктопного приложения.
Мы узнали основы использования конфигурации объектов и я уверен, что все стало понятнее, после того как мы подробно рассмотрели это например приложений. Но самое главное это то, что конфигурация оъекть очень важна когда имеешь дело с **Ext**. Так что чем быстрее вы поймете суть, тем лучше вы будете.
Не беспокойтейсь если вы еще не до конца поняли принципы конфигурации. У нас впереди еще много времени чтобы разобраться с этим. А теперь давайте перейдем к одной из моих любимых тем —формы.