

## Класс Array

Класс Array обеспечивает доступ к массивам и их обработку. Индексы массива начинаются с нуля. Это значит, что первый элемент массива имеет индекс [0], второй элемент - индекс [1] и т. д. Чтобы создать объект Array, можно воспользоваться конструктором new Array(). Объект Array() можно также вызвать в виде функции. Кроме того, можно воспользоваться оператором доступа к массиву ([]), чтобы инициализировать массив или выполнить доступ к его элементам.

В элементе массива можно сохранять широкий спектр типов данных, включая числа, строки, объекты и даже другие массивы. Можно создать **многомерный массив**. Для этого нужно создать индексированный массив и присвоить каждому из его элементов другой индексированный массив. Такой массив считается многомерным, поскольку его можно использовать для представления данных в виде таблицы.

Массивы являются **разреженными**. Это значит, что один элемент может иметь индекс 0, а другой - индекс 5, но позиции индекса между двумя этими элементами будут пустыми. В подобном случае элементы на позициях с 1 по 4 не определены, что указывает на отсутствие элемента, а не обязательно на присутствие элемента со значением undefined.

Присваивание массива выполняется по ссылке, а не по значению. Когда одна переменная массива присваивается другой переменной массива, они обе относятся к одному и тому же массиву:

```
var oneArray:Array = new Array("a", "b", "c");
var twoArray:Array = oneArray; // Both array variables refer to the same array.
twoArray[0] = "z";
trace(oneArray);                // Output: z,b,c.
```

Не применяйте класс Array для создания массивов ассоциативных элементов (называемых также хэшами), которые представляют собой структуры данных, элементам которых присваиваются имена, а не числа. Ассоциативные массивы, называемые также хэш-массивами или просто хэшами, позволяют задавать и хранить значения, с каждым из которых ассоциируется некоторая строка, называемая ключом. В дальнейшем обращение к сохраненному значению осуществляется с помощью его ключа. Хэш-массивы позволяют легко реализовывать динамические структуры данных (списки, деревья и т. д.), а также функциональность простейших систем управления базами данных. Концептуально хэш-массивы представляют собой неупорядоченное множество ассоциированных пар ключ/значение, что отражено в семантике их конструкторов.

Чтобы создать массивы ассоциативных элементов, воспользуйтесь классом Object. Хотя ActionScript позволяет создавать массивы ассоциативных элементов с помощью класса Array, ни один из методов и свойств класса Array нельзя использовать с этими массивами.

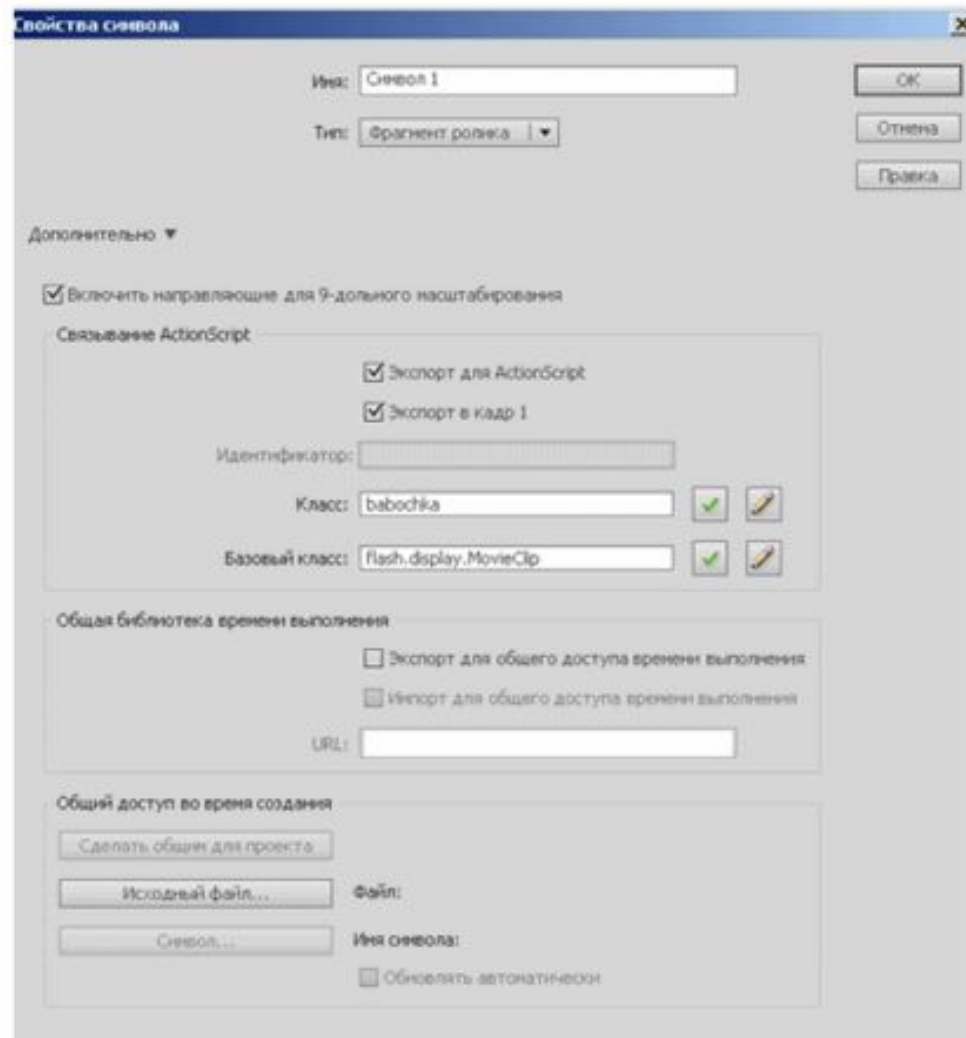
Класс Array можно расширить, переопределив или добавив методы. Однако этот подкласс необходимо сделать динамическим, чтобы не утратить возможность сохранения данных в массиве.

Пример: Создадим программу, в которой клип из библиотеки будет загружаться на монтажный кадр и дублироваться заданное количество раз при помощи массива.

1. Создаем анимированный клип в библиотеке



2. Открываем окно **Свойства символа** в библиотеке, ставим галочку в переключателе Связывание ActionScript и в поле **Класс** вписываем имя babochka.





3. И в первом кадре пишем код:

```
// задаем массив с именем babochkas
```

```
var babochkas = new Array();
```

```
// задаем цикл от 1 до 8 (не включая 8), с шагом 1
```

```
for (var n:uint=1; n<8; n++) {
```

```
// каждому элементу массива (babochkas[1], babochkas[2] ... babochkas[7] присваиваем объект babochka, который создан в библиотеке  
babochkas[n]= new babochka();
```

```
// располагаем элементы массива babochkas[n] на сцене по координате X по порядку с шагом n*65
```

```
babochkas[n].x= n*65;
```

```
// располагаем элементы массива babochkas[n] на сцене по координате Y=75
```

```
babochkas[n].y=75;
```

```
// добавляем элементы массива на сцену методом addChild
```

```
addChild(babochkas[n]);
```

```
}
```

## Объект Array. Методы и свойство

### Array.push()

Синтаксис:

Имя\_массива.push(элемент0, элемент1...)

Этот метод добавляет указанные элементы в конец массива и возвращает новую длину массива.

### Array.splice()

Синтаксис:

Имя\_массива.splice(start,deleteCount, элемент0, элемент1...элементN )

Этот метод удаляет (или добавляет)элементы из массива массива.

start – индекс элемента в массиве, с которого начинается удаление;

deleteCount – количество удаляемых элементов;

элемент0... - значения добавляемых элементов (необязательный аргумент).

### Array.pop()

Синтаксис:

Имя\_массива.pop()

Этот метод применяется для удаления последнего элемента массива и возвращает его значение. При вызове данного метода последний элемент массива удаляется, а значение, находившееся в этом элементе, возвращается.

### Array.shift()

Синтаксис:

Имя\_массива.shift()


Этот метод применяется для удаления первого элемента массива и возвращает его значение.

### Array.unshift()

Синтаксис:

Имя\_массива.unshift(элемент0, элемент1...)

Этот метод добавляет элементы в начало массива и возвращает новую длину массива.



Array.length()

Синтаксис:

Имя\_массива.length

Это свойство объекта Array, которое возвращает длину массива.